

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema Integrado de Gestão de Plataforma de Auditoria Web

Pedro Jorge Santos

Relatório de Projecto realizado no âmbito do Mestrado Integrado em Engenharia
Electrotécnica e de Computadores Major Telecomunicações

Orientador: João Correia Lopes (Prof.)

Co-orientador: Rui Ribeiro (Eng.)

Junho de 2009

Resumo

Actualmente a *Internet* é um meio de comunicação que as empresas utilizam cada vez mais como parte integrante da sua estratégia de *marketing* e publicidade.

A publicidade *online* é actualmente a única fonte de rendimento para muitos sítios *Web*. Empresas como a Google, Yahoo e Microsoft reconhecem o potencial da publicidade *online* e são actualmente consideradas como as principais agências de publicidade *online*. A North America Equity Research [1] estima que a Google irá crescer cerca de 55% este ano e que as suas receitas serão aproximadamente 3 mil milhões de dólares. Devido à crescente importância e à grande quantidade de dinheiro que este tipo de publicidade gera, tem sido alvo de fraudes.

A AuditMark é uma companhia que desenvolve soluções de segurança e auditoria *Web*, ou seja, serviços como auditoria de campanhas de publicidade online e análise de tráfego *Web*. Esta empresa funciona como uma entidade reguladora da publicidade *online*. As actividades da AuditMark exigem a gestão de muita informação. Esta informação deve depois ser disponibilizada aos seus clientes, os anunciantes, para que eles possam gerir da melhor maneira a sua publicidade.

O objectivo deste projecto é desenvolver um sistema de gestão da plataforma de auditoria da AuditMark. Esta será uma ferramenta personalizada para controlar este *workflow* que trará à companhia uma mais valia importante. Esta ferramenta vai permitir que toda a informação de um cliente esteja disponível em qualquer sítio, a qualquer hora, apenas à distância de um clique.

Abstract

Currently the Internet is a way of communication that companies increasingly use as part of its strategy of marketing and advertising.

Online advertising is currently the only source of income for many websites. Companies such as Google, Yahoo and Microsoft recognise the potential of online advertising and are currently considered as the main online advertising agencies. North America Equity Research[1] estimates that Google's growth in 2009 will be around 55%. Due to its increasing importance and the large amount of money that online advertising generates, it has been target of fraud.

AuditMark is a company that develops security solutions and web audit services such as on-line advertising campaigns and Web traffic analysis. This company acts as a regulator of online advertising.

The activities of AuditMark require the management of much information. This information should then be available to customers, advertisers, so they can manage their advertising the best way.

The objective of this project is to develop a web application to control AuditMark's auditing system. This application will be a custom tool to control this workflow. This tool will allow all the information to be available for their clients anywhere, at any time, only at a distance of a click.

Agradecimentos

Aproveito este espaço para agradecer a todos que de alguma forma contribuíram para a concretização e desenvolvimento deste Projecto.

Agradeço ao Professor João Correia Lopes, todas as suas sugestões e apreciações feitas ao longo do semestre. Ao Eng. Rui Ribeiro a oportunidade que me deu para integrar o mundo do trabalho, assim como todo o apoio que me deu ao longo deste semestre.

A todos eles, o meu Obrigado.

O Autor

*“When I took office, only high energy physicists
had ever heard of what is called the Worldwide Web....
Now even my cat has its own page.”*

Bill Clinton

*“I love deadlines.
I like the whooshing sound they make as they fly by.”*

Douglas Adams

*“A significant part of the distribution of goods and services in this country is going to move from
conventional channels to some sort of Internet system — whether it’s retail goods or services.”*

Alan Greenspan

Conteúdo

1	Introdução	1
1.1	Enquadramento e Motivação	1
1.2	Objectivos	3
1.3	Estrutura do documento	4
2	Auditoria e Análise de Tráfego Web	5
2.1	Soluções Existentes	5
2.1.1	Google Analytics	6
2.1.2	WebTrends	6
2.1.3	Omniure SiteCatalyst	7
2.1.4	ClickTracks	8
2.1.5	NetTracker	9
2.1.6	Urchin	10
2.1.7	ClickForensics	11
2.2	Tecnologias de Apoio ao Desenvolvimento	12
2.3	Conclusão	14
3	Ferramentas para Desenvolvimento Web	17
3.1	Base de Dados	17
3.2	Framework Javascript	18
3.3	Sistema de Renderização de Gráficos	19
3.3.1	AnyChart	19
3.3.2	Flash Charts Pro	20
3.3.3	amCharts	20
3.3.4	Fusion Charts	20
3.3.5	Emprise Java Script Charts	20
3.3.6	Swiff Chart	20
3.3.7	XML/SWF	21
3.3.8	Conclusão	21
3.4	Framework de apoio ao desenvolvimento	21
3.5	Testes de frameworks	25
4	Especificação de Requisitos	27
4.1	Site Corporativo / Registo e login	27
4.2	Acessos e permissões	28
4.3	Interface de análise	28
4.4	Documentação e suporte	29
4.5	Casos de Utilização	29

5	Implementação	35
5.1	Arquitetura do Sistema	35
5.1.1	Arquitetura Física	35
5.1.2	Arquitetura Lógica	36
5.2	Modelo de Dados	37
5.3	Esboço da Interface Gráfica	37
5.4	Módulos	39
5.4.1	Sítio Web	39
5.4.2	Área de Clientes	42
5.4.3	Interface de Administração	43
5.4.4	Interface de Análise	47
5.4.5	Sistema de Ajuda	53
5.5	Testes	55
5.5.1	Teste de Implementação	55
5.5.2	Testes de Segurança	56
5.5.3	Teste de Validação	56
6	Conclusões e Trabalho Futuro	59
	Referências	61
A	Análise dos Servidores de Anúncios	65

Lista de Figuras

2.1	Interface do Google Analytics	7
2.2	Interface do WebTrends	8
2.3	Interface do Omniture	9
2.4	Interface do ClickTracks	10
2.5	Interface do NetTracker	11
2.6	Interface do Urchin	12
2.7	Interface do ClickForensics	13
2.8	Modelo MVC	13
4.1	Diagrama de Pacotes	30
4.2	Casos de utilização — Site Corporativo	30
4.3	Casos de utilização — Sistema de Ajuda	31
4.4	Casos de utilização — Área de Cliente	31
4.5	Casos de utilização — Interface de Análise	32
4.6	Casos de utilização — Interface de Administração	33
5.1	Arquitectura Física	35
5.2	Modelo de Dados	38
5.3	Esboço do Site Corporativo	39
5.4	Esboço da Interface de Administração	40
5.5	Esboço do Dashboard de Campanhas	40
5.6	Esboço do Dashboard de Relatórios	41
5.7	Formulário de Registo	42
5.8	Formulário de Recuperação Username/Password	43
5.9	Menu Lateral da Interface de Administração	44
5.10	Formulário de Edição dos Dados Pessoais	45
5.11	Tabelas para Gestão de Contas e Campanhas	46
5.12	Tabelas para Gestão de Permissões	46
5.13	Tabelas para Visualizar Facturas	47
5.14	Menu para personalizar gráficos e tabelas	48
5.15	Selector de Campanhas	48
5.16	Gráfico com Método Comparativo	49
5.17	Gráfico com Método Somatório	49
5.18	Selector de Series	50
5.19	Botão de Validação	50
5.20	Dashboard Relatórios	51
5.21	Mover Widgets	51
5.22	Configuração de Widgets	52

5.23	Gráfico de Barras — Modo Soma	53
5.24	Gráfico de Barras — Modo Comparativo	54
5.25	Gráfico Tarte	54

Lista de Tabelas

3.1	Primeira fase de comparação de frameworks	24
3.2	Primeira fase de comparação de frameworks(Cont.)	25
3.3	Segunda fase de comparação de frameworks	25
5.1	Vulnerabilidades encontradas e respectiva solução	56
A.1	Atributos da Conta	66
A.2	Atributos da Campanha	66
A.3	Atributos do Grupo de Anúncios	66
A.4	Atributos do Anúncio	67

Abreviaturas e Símbolos

Screenshot	<i>Captura do ecrã</i>
Browser	<i>Navegador web</i>
Cookie	<i>Pequeno texto armazenado no computador de um utilizador por um browser</i>
CSS	<i>Cascading Style Sheets</i>
MVC	<i>Model-view-controller</i>
AJAX	<i>Asynchronous Javascript And XML</i>
WWW	<i>World Wide Web</i>
HTTP	<i>Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)</i>
HTML	<i>HyperText Markup Language (Linguagem de Marcação de Hipertexto)</i>
HIT	<i>Pedido que é feito a um servidor HTTP por um determinado arquivo (imagem, HTML, Javascript, CSS, etc)</i>
ROI	<i>Return On Investment (Retorno Sobre Investimento)</i>
SaaS	<i>Software as a Service (Software como um serviço)</i>
Data Mining	<i>Processo de extrair padrões a partir de dados ocultos</i>
Click Fraud	<i>Actividade ilegítima que consiste em produzir cliques (i.e. pedidos HTTP) de forma automática ou manual, em pequena ou larga escala, no sentido de obter um proveito ou prejudicar um terceiro.</i>
PPC	<i>Pay Per Click (Pagar por clique)</i>

Capítulo 1

Introdução

Este capítulo descreve o projecto efectuado na AuditMark, no âmbito da disciplina Dissertação do curso de Engenharia Electrotécnica e de Computadores leccionado na Faculdade de Engenharia da Universidade do Porto (FEUP). O projecto teve como orientadores o Professor João Correia Lopes e, por parte da AuditMark, o Eng. Rui Ribeiro. Este capítulo descreve o contexto em que se insere este projecto, motivação, objectivos e a organização deste documento.

1.1 Enquadramento e Motivação

Nos últimos anos, tornou-se visível o entusiasmo do mercado publicitário em relação à *Internet* como um meio repleto de novas potencialidades para a difusão das suas mensagens. A atribuição de recursos que, tradicionalmente, eram destinados à televisão, ao rádio e aos jornais, entre outros, assim como o aumento da facturação da área, ano a ano, são indícios do prestígio que este novo media conquistou. Ao mesmo tempo, o interesse da indústria da publicidade também é um indicador da sua importância e desenvolvimento, enquanto meio de comunicação.

Actualmente as empresas utilizam cada vez mais a *Internet* como parte integrante da sua estratégia de *marketing*. Estar *online* é fácil. Para tal basta adquirir um domínio, construir um sítio *Web*, colocá-lo *online* e a sua empresa está na *World Wide Web*. Se estar *online* é fácil, saber como estar já é um assunto completamente diferente. O saber estar implica utilizar as várias tecnologias disponíveis de acordo com os objectivos da empresa, em vez de o fazer como se o sítio *Web* fosse algo independente e sem qualquer interligação com os restantes canais de publicidade.

A publicidade existe para aumentar as receitas e as vendas de uma empresa. Mas a publicidade é em si uma tarefa dispendiosa, que envolve uma grande quantidade de dinheiro e é sempre um grande risco, uma vez que as suas consequências não são conhecidas. Para diminuir este risco, as empresas tentam anunciar através de revistas, jornais, e publicações, bem como a *Internet*. A publicidade online é eficaz porque permite uma resposta directa que é benéfica tanto para o consumidor como para o vendedor. O consumidor apenas tem de clicar num anúncio para saber

mais sobre um produto e, se assim o desejar, pode comprar o produto online sem ter de se deslocar. Ao ocupar o seu lugar na *Web*, esta ferramenta de *marketing* de carácter persuasivo encontra, à sua disposição, uma série de novos espaços a serem explorados, cada um oferecendo funcionalidades diferenciadas, capazes de afectar a forma como a publicidade é produzida, distribuída e recebida.

A publicidade *online* é actualmente a única fonte de rendimento para muitos sítios *Web*. Empresas como a Google, Yahoo e Microsoft reconhecem o potencial deste tipo de publicidade e são actualmente consideradas como as principais agências de publicidade *online*. Uma agência de publicidade *online* tem como função disponibilizar anúncios para sítios *Web* que se dedicam à sua exposição. A cada clique efectuado num anúncio, a empresa responsável pelo anúncio (o anunciante) paga uma quantia que é depois repartida entre a agência de publicidade e o sítio da *Web* que expôs o anúncio. A este modelo de tarifação dá-se o nome de PPC ¹.

Devido à crescente importância deste tipo de publicidade e à grande quantidade de dinheiro que gera, tem sido alvo do que se designa habitualmente por *Click Fraud* ². Esta situação fez com que as entidades envolvidas neste negócio desenvolvessem uma maneira de identificar possíveis situações de fraude. É assim que surgem no mercado as ferramentas de Auditoria *Web*, como resposta a um problema que à muito pedia uma solução. Estas ferramentas têm como função analisar todo o processo e disponibilizar às partes envolvidas todo um conjunto de informação que as ajude a identificar possíveis situações de fraude.

O acesso à informação de qualidade é essencial para que uma empresa alcance os seus objectivos. A importância da qualidade da informação e a maneira como a informação é disponibilizada são alguns dos factores responsáveis pelo crescimento e constante uso da *Internet* pelo mercado de tecnologia da informação. Com esse mercado crescente, as organizações começaram a desenvolver sistemas de informação para a *Web*, onde a usabilidade da interface deve ser levada em consideração em todas as interacções dos sistemas que lidam com informações, principalmente para aqueles sistemas nos quais o utilizador interage directamente com o sítio *Web*.

É através da interface que os utilizadores têm acesso às funções da aplicação. Com isso, factores de medidas de usabilidade como eficácia, eficiência, satisfação, segurança, tempo de aprendizagem e retorno de investimento passam a depender de um bom projecto de interface gráfica. Na visão do utilizador, um produto com qualidade é aquele que responde às suas necessidades, sendo fácil de usar, evitando ideias e soluções equivocadas, concentrando recursos em actividades organizadas através de uma metodologia específica. Portanto, para garantir uma interface adequada às necessidades do utilizador, alguns factores básicos como informações precisas, planeamento correcto, projecto adequado, execução supervisionada e uma constante avaliação dos resultados obtidos são fundamentais. Essas adequações do produto podem ser garantidas através de um processo de construção organizado e através da avaliação da interface.

A AuditMark é uma empresa que desenvolve soluções de Auditoria *Web*, ou seja, serviços como auditoria de campanhas de publicidade e análise de tráfego *Web*. Esta empresa funciona

¹Pay Per Click

²Actividade ilegítima que consiste em produzir cliques (i.e. pedidos HTTP ³) de forma automática ou manual, em pequena ou larga escala, no sentido de obter um proveito ou prejudicar um terceiro.

como uma entidade reguladora da publicidade *online*, identificando possíveis situações de *Click Fraud*. As actividades da AuditMark exigem a gestão de muita informação. Esta informação é recolhida e armazenada em base de dados para depois ser analisada por ferramentas personalizadas. Os resultados desta análise devem depois ser disponibilizados aos seus clientes, os anunciantes, para que eles possam gerir da melhor maneira a sua publicidade.

O uso de uma ferramenta personalizada com uma interface fácil de utilizar para controlar o *workflow* da empresa trará uma mais valia importante. Esta ferramenta vai permitir que toda a informação de um cliente esteja disponível em qualquer sítio, a qualquer hora, apenas à distância de um clique. Os clientes vão poder associar, gerir e analisar todas as suas campanhas publicitárias a partir de um único sítio.

Através desta interface, os clientes vão ter acesso a todo um conjunto de informação, apresentada de uma forma simples e intuitiva através de diferentes tipos de gráficos e tabelas, assim como um conjunto de funcionalidades que lhes vai permitir comparar o desempenho das suas campanhas publicitárias e assim tomar as melhores decisões.

Foi a partir destes pressupostos que nasceu a motivação para realizar este projecto.

1.2 Objectivos

O objectivo deste projecto é desenvolver um sistema de gestão da plataforma de auditoria *Web* da AuditMark. Este sistema deve oferecer todo o sistema básico padrão de uma aplicação *Web*, ou seja, controlo de acessos, acesso à base de dados, gestão de conteúdos e uma interface fácil de usar, assim como um conjunto de funcionalidades personalizadas:

- Permitir aos utilizadores efectuar registo/login usando uma identidade *Web* como o OpenId⁴;
- Permitir aos utilizadores importar as suas campanhas publicitárias a partir de *Ad Servers*⁵ como *OpenX* ou *Google Adwords*;
- Possibilidade de visualizar e analisar todas as características do tráfego gerado pelas suas campanhas publicitárias;
- Possibilidade de visualizar a mesma informação em forma de gráfico ou tabela, podendo escolher o intervalo temporal;
- Possibilidade de analisar individualmente o desempenho de cada campanha ou comparar o desempenho de várias campanhas;
- Possibilidade de partilhar informação entre vários utilizadores podendo atribuir diferentes permissões a cada um deles;

⁴OpenId é um sistema de identificação; trata-se de uma rede distribuída na qual a identidade do utilizador é dada através de um *URL* ou *XRI* que pode ser verificada por qualquer servidor executando o protocolo.

⁵Servidor de anúncios — Guarda anúncios que são utilizados em publicidade *online* e entrega-os aos visitantes.

- Possibilidade de exportar toda a informação disponível em diversos formatos de ficheiro como **XLS**⁶, **CVS**⁷ ou **XML**⁸.

Este sistema deve estar preparado para interagir com o *framework* que fornece os serviços de auditoria da AuditMark e deve ser compatível com os *browsers* dominantes: Internet Explorer (versão 6 e posteriores) e Mozilla Firefox (versão 2 e posteriores). Também faz parte dos objetivos integrar um sistema de Ajuda para responder às questões dos utilizadores e disponibilizar o sistema em duas línguas: Português e Inglês.

1.3 Estrutura do documento

Para além da introdução, esta dissertação contém mais 5 capítulos. No capítulo 2 são descritas as soluções que existem actualmente no mercado para realizar auditoria e análise de tráfego *Web* assim como as tecnologias mais usadas em desenvolvimento *Web*. No capítulo 3 são apresentadas as ferramentas mais usadas para desenvolvimento *Web*. No capítulo 4 são descritos os requisitos deste projecto. No capítulo 5 é descrita a solução encontrada e os detalhes de implementação deste projecto. No capítulo 6 é feita uma breve reflexão sobre o trabalho realizado e trabalho futuro.

⁶Formato utilizado pelo Microsoft Excel.

⁷*Comma Separated Values*

⁸eXtensible Markup Language

Capítulo 2

Auditoria e Análise de Tráfego Web

Este capítulo descreve as soluções que existem actualmente no mercado para realizar auditoria e análise de tráfego *Web* assim como as tecnologias mais usadas em desenvolvimento *Web*. A análise de soluções semelhantes vai ser focada na sua interface *Web* e funcionalidades disponibilizadas para, assim, podermos compreender o que está feito e como podemos fazer melhor. Vamos também fazer um levantamento das tecnologias que, hoje em dia, são mais utilizadas em desenvolvimento de aplicações para a *Web*.

2.1 Soluções Existentes

A AuditMark é uma companhia que desenvolve soluções de análise de tráfego e auditoria *Web*. Para desenvolver uma interface *Web* que melhor se adapte às necessidades dos utilizadores foi necessário compreender como funcionam estas ferramentas. De entre as soluções disponíveis, decidimos destacar as seguintes devido à sua popularidade, usabilidade da interface gráfica e conjunto de funcionalidades disponibilizadas:

- Google Analytics [2]
- WebTrends [3]
- Omniture SiteCatalyst [4]
- Lyris ClickTracks [5]
- NetTracker [6]
- Urchin [7]
- ClickForensics [8]

Nas subsecções seguintes vamos analisar cada uma destas ferramentas.

2.1.1 Google Analytics

Google Analytics é uma ferramenta gratuita disponibilizada pela Google que funciona em conjunto com o Google Adwords. Ao activar o serviço o utilizador recebe um pedaço de código Javascript que deve ser inserido nas páginas que deseja analisar. A cada exibição dessa página, são enviadas ao sistema da Google estatísticas de visitação que são posteriormente disponibilizadas ao utilizador.

Um dos aspectos bons desta ferramenta é o facto de ser realmente muito simples de configurar. O Google Analytics começa a coleccionar dados dos visitantes poucos minutos após a conta estar configurada. Esta ferramenta dispõe de mais de 80 diferentes tipos de relatórios com os quais se pode analisar o que está a acontecer com o nosso sítio *Web*. Um dos aspectos negativos é o facto da toda a análise estar vinculada às campanhas do AdWords.

O Google Analytics tem uma interface simples e limpa com integração de AJAX, na qual é muito fácil navegar entre relatórios. Quando se inicia uma sessão é imediatamente apresentada uma panorâmica colorida e fácil de ler que mostra gráficos com informação sobre os visitantes. Esta ferramenta tem toda a navegação concentrada num menu lateral do lado esquerdo, na qual os diferentes tipos de relatórios são agrupados em árvore. O selector de intervalos de tempo é simples de usar e torna muito fácil escolher um determinado dia/semana/mês, ou escolher um intervalo personalizado.

O Google Analytics é capaz de identificar, além da tradicional taxa de exibição e *hit* de uma página, a localização geográfica do visitante e a forma como chegou à página, quer seja através de *links* de outros sítios *Web*, motor de busca, através do AdSense ou mesmo directamente pelo endereço.

Consegue também identificar o sistema operativo e *browser* com as respectivas versões, a resolução do ecrã do utilizador e a presença de diferentes *plugins* tais como Javascript, reproduutor de Flash, entre outros. Todas estas análises podem ser disponibilizadas em períodos diários, semanais, mensais ou mesmo anuais. Apesar de todos os relatórios, gráficos e imagens serem visualizados directamente através do sítio *Web* do Google, também tem a opção de fazer *download* dos dados em XML, CSV ou formatos de texto — o que é interessante pois permite guardar os dados mais importantes. A Figura 2.1 mostra um exemplo da interface do Google Analytics.

2.1.2 WebTrends

Alugar ou comprar é uma das grandes questões filosóficas da civilização moderna: o WebTrends dá-lhe as duas opções. Está disponível como um pacote de software que se pode instalar no nosso próprio servidor ou como um serviço *on-demand* hospedado. Esta ferramenta tem uma interface altamente visual e intuitiva, em que os seus gráficos coloridos simplificam até a mais complexa das análises.

A finalidade desta ferramenta é disponibilizar um conjunto de dados que são coleccionados quando um utilizador acede a um determinado sítio *Web*. Uma das grandes vantagens é a possibilidade de apresentar as informações de forma compreensível para o utilizador final. Para que

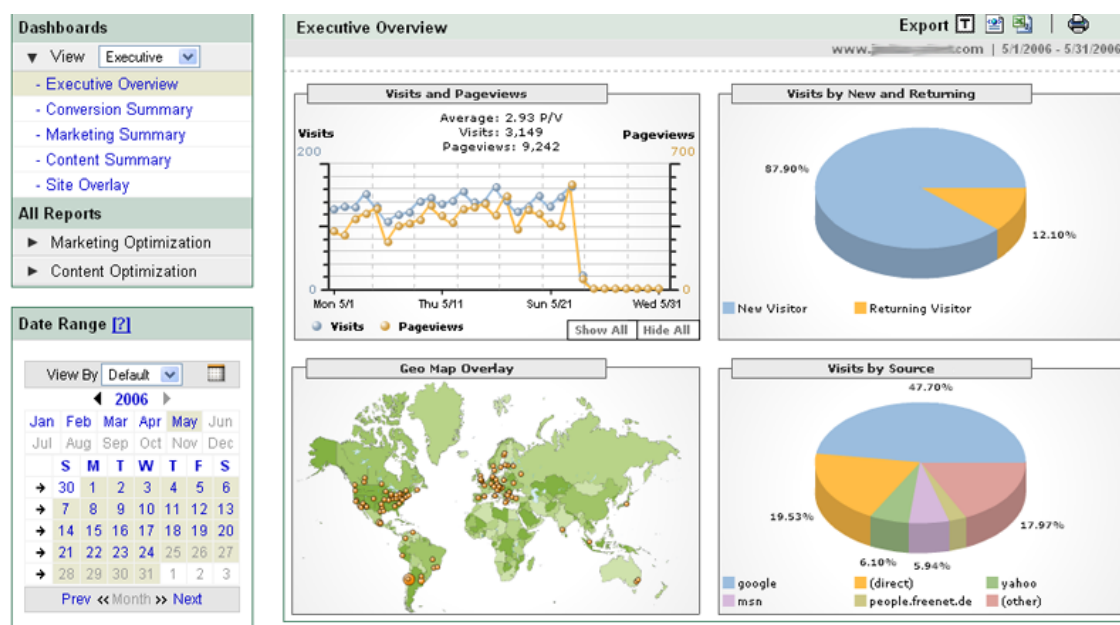


Figura 2.1: Interface do Google Analytics

isso aconteça existe uma interface *Web* altamente configurável, onde é possível processar, alterar e personalizar toda a informação.

Após a análise dos dados colecionados, oferecem informações como *top pages*, *referrers*, palavras chave, informações sobre os visitantes, etc. Toda esta informação é disponibilizada em forma de tabelas altamente detalhadas com opções *drill-down* ou em forma de gráficos. O utilizador pode especificar qual o intervalo de tempo que deseja visualizar, a ferramenta garante pelo menos 3 meses de informação disponível. É também possível exportar toda a informação em diferentes formatos como *MS Word*, *Excel*, *CSV* ou *PDF*.

Um dos recursos mais interessantes é o *SmartView* que permite isolar um elemento específico numa página e ver a percentagem de visitantes que clicaram nesse elemento, o número de cliques (com cores codificadas por popularidade) e o caminho que levou a maioria dos visitantes a uma determinada página. Isto é muito útil para determinar a eficácia dos *links* e a maneira como eles influenciam a navegação e a quantidade de tempo que cada utilizador gasta em cada página.

WebTrends também se integra bem com o Microsoft Excel, permitindo exportar informação directamente para o Excel e criar relatórios conforme o desejado. Esta funcionalidade permite efectuar análises mais complexas do que é possível fazer dentro do WebTrends. A Figura 2.2 mostra um exemplo da interface do Webtrends.

2.1.3 Omniture SiteCatalyst

Omniture SiteCatalyst é uma ferramenta que disponibiliza serviços de optimização de sítios *Web* e palavras-chave, assim como variadas análises. O grande objectivo desta ferramenta é garantir o melhor e mais eficiente *ROI* para que profissionais de *marketing* tenham o melhor de-

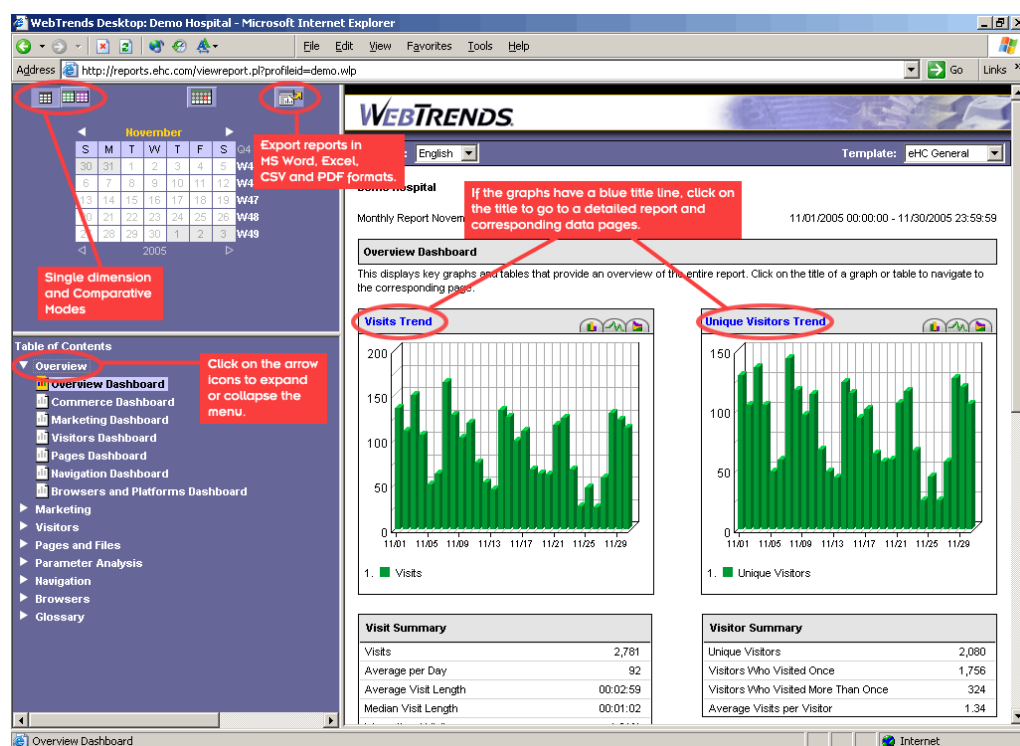


Figura 2.2: Interface do WebTrends

sempenho no mundo digital, aliando os objectivos da empresa com o do consumidor otimizando investimentos.

Omniture SiteCatalyst disponibiliza diversas funcionalidades tais como dados relacionados com a navegação, comércio electrónico, conteúdo e outras informações detalhadas do visitante. Tem também um interessante painel de instrumentos. Uma das grandes funcionalidades que oferece é a comparação, o que permite comparar as diferentes campanhas, de modo que as campanhas possam ser optimizadas.

Esta ferramenta tem uma interface com boa navegabilidade, apresentação legível e de fácil compreensão. Apresenta os relatórios em duas formas combinadas, gráficos e tabelas, o que ajuda a melhor compreender a informação que estamos a visualizar. Um dos recursos mais interessantes é a capacidade de comparar informação relativa a intervalos de tempo diferentes, aliado a um sistema de filtragem com um conjunto bastante completo de expressões lógicas, o que a torna numa ferramenta muito útil. A Figura 2.3 mostra um exemplo da interface do Omniture SiteCatalyst.

2.1.4 ClickTracks

Bem conhecida na indústria pela sua interface fácil de usar e um poderoso sistema de relatórios, ClickTracks é uma ferramenta poderosa para análise do comportamento dos visitantes numa *sítio Web*. Os utilizadores podem visualizar as suas páginas exactamente como elas são mostradas aos visitantes, assim como estatísticas de número de clicks em cada anúncio.

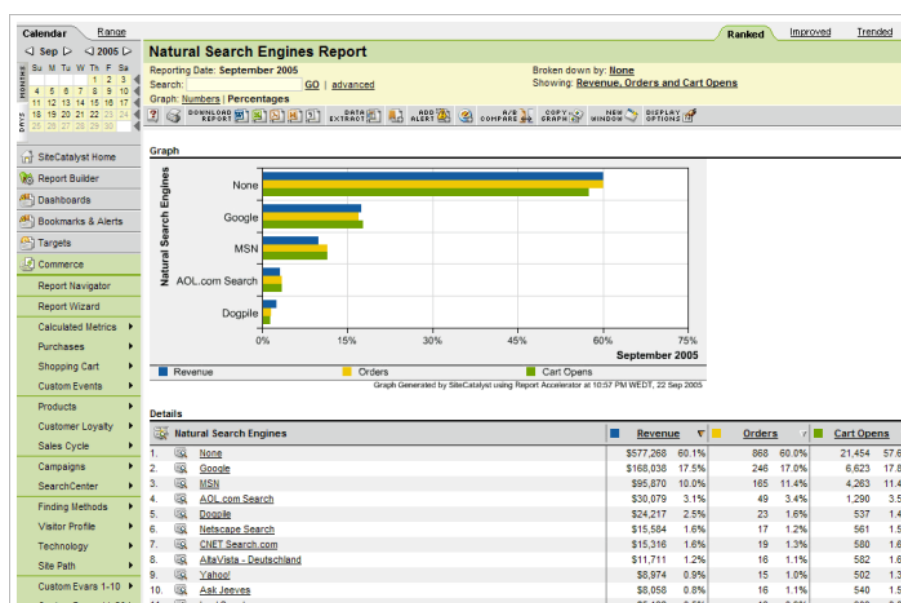


Figura 2.3: Interface do Omniture

ClickTracks Optimizer oferece uma variedade de relatórios e funcionalidades como a hipótese de comparar diferentes versões das páginas para assim se decidir pela melhor versão, análises sobre a popularidade, tempo gasto e número de visitas de cada página, assim como estatísticas de palavras-chave que levaram o visitante aquela página.

O utilizador pode personalizar as estatísticas que são disponibilizadas na análise das páginas. Esta ferramenta disponibiliza também funcionalidades para analisar o comportamento e desempenho das palavras-chave utilizadas, permitindo importar dados de *Ad Servers* externos e assim analisar o *ROI*¹, assim como funcionalidades para detectar *Click Fraud*, analisando o desempenho das campanhas. Toda a informação disponibilizada pode ser exportada para ficheiro em diferentes formatos.

A característica mais interessante desta ferramenta é a sua interface gráfica que possui um menu simples, que permite que ao utilizador decidir que tipo de relatório deseja visualizar, juntamente com o intervalo de datas a utilizar. ClickTracks também permite aos utilizadores importar ficheiros a partir de um servidor remoto usando o *ClickTracks' FTP*, ou importá-los a partir de um ficheiro local. A Figura 2.4 mostra um exemplo da interface do ClickTracks.

2.1.5 NetTracker

NetTracker é uma ferramenta de análise de tráfego *Web* que fornece uma análise detalhada sobre o comportamento dos visitantes. Esta análise inclui identificação dos visitantes que abandonam uma página, determinação da taxa de visualização de uma página e estatísticas sobre o comportamento dos visitantes através de padrões de compra, rentabilidade e demografia.

¹ Return On Investment (Retorno Sobre Investimento)

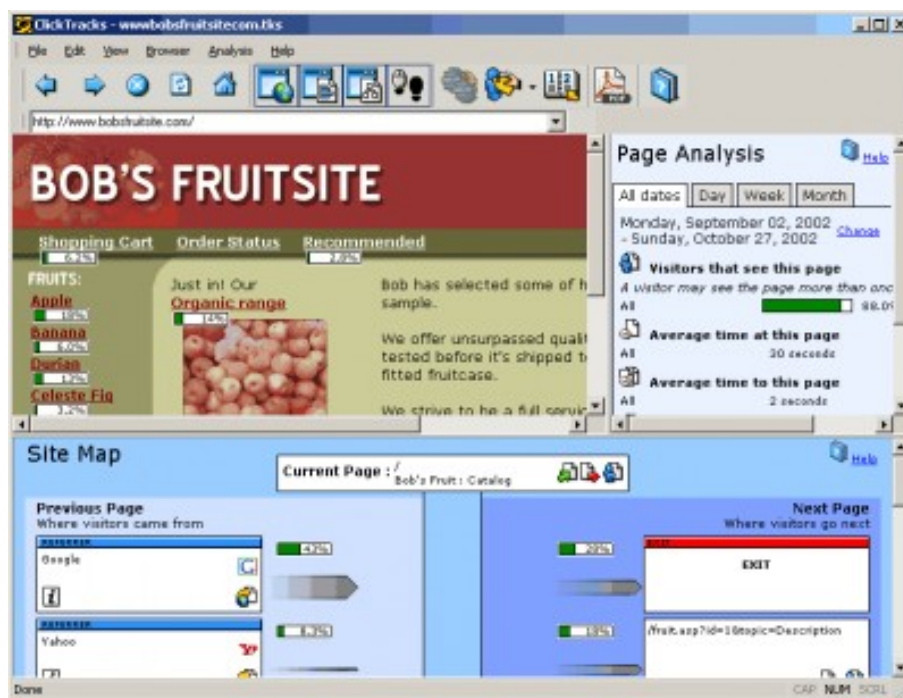


Figura 2.4: Interface do ClickTracks

Esta ferramenta permite identificar *links* danificados, assegurar que as páginas são otimizadas para visualização nos *browsers* usados pela maior percentagem dos utilizadores, identificar páginas lentas, analisar o desempenho dos servidores, visualizar visitas por dia, a fim de analisar os resultados das campanhas publicitárias, analisar o tempo que cada visitante passa em cada página, de modo a determinar tendências no uso do sítio *Web*, ou mesmo analisar a largura de banda utilizada, para garantir que os visitantes podem aceder ao sítio *Web* sem problemas. A Figura 2.5 mostra um exemplo da interface do NetTracker.

2.1.6 Urchin

Urchin é uma ferramenta de análise de tráfego *Web* concebida para analisar o tráfego para um ou mais sítios *Web* e fornecer uma análise personalizada, detalhada, útil e fácil de compreender. Esta ferramenta utiliza a tecnologia Adobe SVG para exibir gráficos e imagens que carregam quase instantaneamente. Permite alterar o intervalo de datas de um relatório e usar filtros para modificar a informação disponibilizada.

Urchin recolhe informações a partir de arquivos de *log* e informações sobre *cookies*. Esta ferramenta permite copiar facilmente qualquer gráfico para *PowerPoint*, *Word*, etc. Como utiliza gráficos baseados em *SVG* é muito rápido a carregar. Entre outras funcionalidades, esta ferramenta disponibiliza *backups* automáticos, gestão de utilizadores, grupos e permissões, gestão e visualização de relatórios em qualquer computador, relatórios em multi-língua, compatibilidade com diversos sistemas operativos como Windows 2003/XP/2000/NT, Mac OS X, UNIX / Linux,



Figura 2.5: Interface do NetTracker

entre outros, exportação de qualquer relatório para o *Excel*, *Word*, ou *TSV*, sistema de ajuda integrado e a possibilidade de comparar relatórios em tempo real. A Figura 2.6 mostra um exemplo da interface do Urchin.

2.1.7 ClickForensics

O modelo de pagamento que se usa actualmente nas campanhas de publicidade online é o modelo Pay Per Click (PPC), ou seja, os anunciantes pagam um determinado valor por cada clique que receberem nos seus anúncios. É importante então saber se cliques sobre um determinado anúncio são ou não falsos. Para isso, os anunciantes ou sítios *Web* expositores, recorrem a ferramentas como esta.

O ClickForensics é um *SaaS*² que permite fazer auditoria *Web*, ou seja, determinar se os cliques gerados sobre um determinado anúncio são ou não fidedignos. Os atributos dos cliques nos anúncios são examinados através de ficheiros e analisados utilizando técnicas heurísticas comprovadas para a determinação da qualidade.

O cérebro de cada produto da ClickForensics é o motor PPC. Este analisa os dados do tráfego utilizando técnicas de *Data Mining*³ de modo a atribuir um factor de qualidade a cada clique e cada visitante. Esta análise examina três categorias de dados:

²Software as a Service (Software como um serviço)

³Processo de extrair padrões a partir de dados ocultos.



Figura 2.6: Interface do Urchin

- Atributos — endereço *IP*, servidor *proxy*, características do *bot*, geo-localização, características dos cabeçalhos *HTTP*, perfil do visitante e características do custo por clique;
- Comportamento — o tempo gasto no site, visualização de padrões de comportamento e tempo entre os cliques;
- Comunidade — comparativo de dados entre toda a comunidade de utilizadores.

Esta é uma das ferramentas mais completas que se analisou, pois enquanto a maior parte se foca apenas na análise de tráfego, o ClickForensics disponibiliza também um vasto conjunto de funcionalidades que permitem identificar situações de *Click Fraud* — o que a maioria das ferramentas não disponibiliza. A Figura 2.7 mostra um exemplo da interface do ClickForensics.

2.2 Tecnologias de Apoio ao Desenvolvimento

Com o aumento da complexidade das aplicações *Web* torna-se fundamental a separação entre os dados e o *layout*. Desta forma, alterações feitas no *layout* não afectam a manipulação de dados, e estes poderão ser reorganizados sem alterar o *layout*. Uma maneira de o conseguir passa pela utilização do modelo de separação que tem o nome de *MVC* — *Model View Controller* (ver Figura 2.8).

O MVC resolve este problema através da separação das tarefas de acesso aos dados, lógica de apresentação e de interacção com o utilizador, introduzindo um componente entre os dois: o *Controller*.



Figura 2.7: Interface do ClickForensics

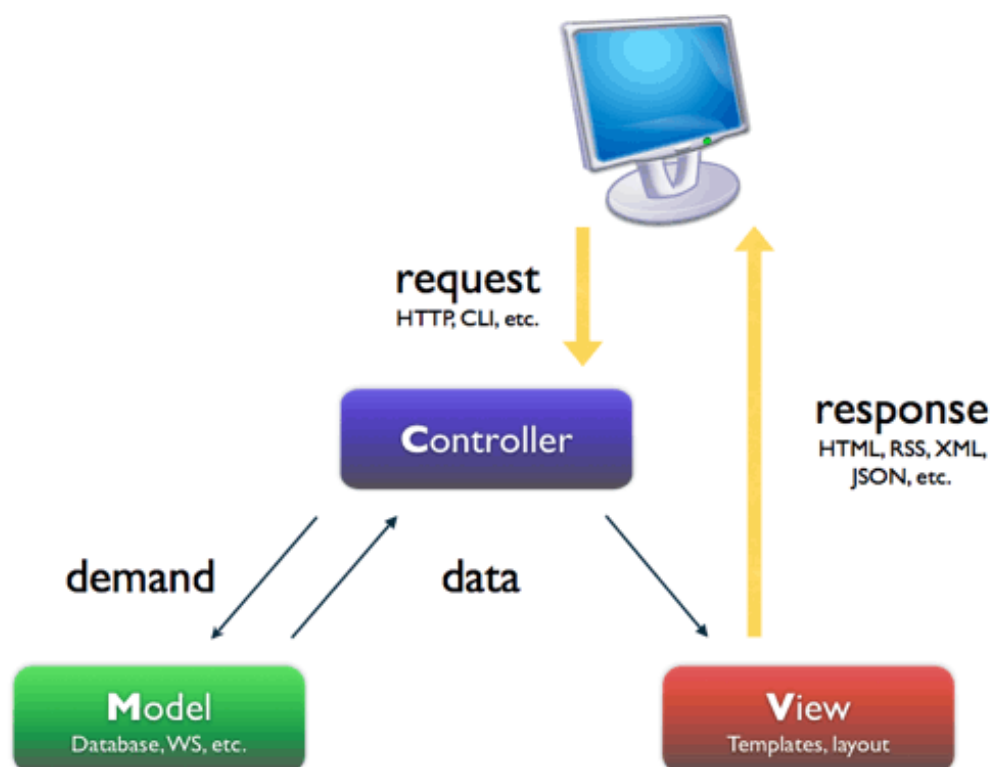


Figura 2.8: Modelo MVC

Em desenvolvimento *Web a View* é a página *HTML* [9] e o código que gera os dados dinâmicos para dentro do *HTML* é o *Controller*. O *Model* é representado pelo conteúdo armazenado em base de dados.

Na *View*, para facilitar a escalabilidade e enriquecer o *layout* o programador deve tirar partido de técnicas como *XML* [10], *CSS* [11], *Javascript* [12], *AJAX* e *Template Engine*.

Uma definição possível para *XML* é a seguinte [13]:

“XML (eXtensible Markup Language) é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais.”

Uma definição possível para *CSS* é a seguinte [14]:

“Cascading Style Sheets (ou simplesmente CSS) é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos numa linguagem de marcação, como HTML ou XML. O seu principal benefício é garantir a separação entre o formato e o conteúdo de um documento.”

CSS permite-nos colocar toda a formatação do documento numa página à parte e criar apenas um *link* para essa página que contém os estilos. Desta forma quando for necessário alterar a aparência do sítio *Web* basta modificar apenas os ficheiros CSS.

Uma definição possível para *AJAX* é a seguinte [15]:

“AJAX (acrónimo em língua inglesa de Asynchronous Javascript And XML) é o uso metodológico de tecnologias como Javascript e XML, providas por navegadores, para tornar as páginas mais interactivas com o utilizador, utilizando-se solicitações assíncronas de informações.”

Normalmente, quando submetemos um formulário numa sítio *Web*, toda a página é recarregada com a nova informação enviada pelo servidor. O uso de *AJAX* permite-nos indicar ao *browser* que queremos recarregar apenas uma parte da página. Desta maneira a navegação torna-se muito mais rápida, visto que não é necessário recarregar a página completa e diminui a largura de banda utilizada.

Javascript é uma linguagem desenvolvida pela Netscape que pode funcionar tanto como uma linguagem orientada a procedimentos ou objectos. Esta técnica permite-nos tornar as páginas *Web* mais dinâmicas.

Um *template web* é uma ferramenta usada para separar os conteúdos da apresentação de uma página *web*. Um *Template Engine* processa *templates web* e procede à sua apresentação.

2.3 Conclusão

Grande parte do sucesso que ferramentas como Google Analytics ou Webtrends alcançaram no mercado, deve-se ao facto de disponibilizarem um vasto conjunto de funcionalidades aliado a uma interface dinâmica e intuitiva. Para isso contribui o uso de gráficos e tabelas interactivas, que ajudam a tornar a interface mais leve e mais fácil de usar.

Esta análise ajudou a compreender que o sucesso de uma aplicação *Web* não é só medido pelo conjunto de funcionalidades disponibilizadas, mas também pela usabilidade da sua interface. Para nos ajudar a desenvolver uma interface completa mas ao mesmo tempo fácil de usar, devemos tirar partido de várias tecnologias como o *Javascript*, *CSS* e *AJAX*.

Após termos efectuado um levantamento das diferentes soluções disponíveis e tecnologias mais usadas, vamos no próximo capítulo descrever o conjunto de ferramentas de apoio ao desenvolvimento disponíveis para criar este tipo de aplicações.

Capítulo 3

Ferramentas para Desenvolvimento Web

Este capítulo descreve as diversas ferramentas existentes para efectuar desenvolvimento *Web*. Vamos analisar as soluções disponíveis para a base de dados, framework *Javascript*, sistema de renderização de gráficos e *framework* de apoio ao desenvolvimento.

3.1 Base de Dados

As bases de dados são o método de armazenamento preferido para ambientes multi-utilizador nos quais é necessário garantir coordenação e coerência de grandes quantidades de informação, para além de um bom desempenho.

Existem actualmente diversas base de dados relacionais disponibilizadas em software livre. O MySQL [16], Oracle Lite [17] e o PostgreSQL [18] são apenas alguns exemplos. Para este projecto ficou decidido que se vai usar PostgreSQL. Uma das razões desta escolha é o facto de esta ser a base de dados seleccionada para todos os projectos na empresa, sendo que a AuditMark tem experiência e várias utilizações da mesma. Como a interface a desenvolver se vai integrar na framework da AuditMark esta vai ter que usar os recursos disponibilizados pela empresa, nomeadamente a base de dados que é PostgreSQL.

PostgreSQL é um sistema para gerir bases de dados e objectos relacionais (SGBDOR) desenvolvido como projecto software livre. Hoje em dia o PostgreSQL é um dos SGBDs ¹ de código aberto mais avançados, contando com recursos como:

- Consultas complexas à base de dados;
- Chaves estrangeiras;

¹Sistema de Gestão de Bancos de Dados

- Integridade transaccional;
- Controle de concorrência multi-versão;
- Suporte ao modelo híbrido objecto-relacional;
- Gatilhos;
- Vistas externas na base de dados;
- Procedimentos armazenados em várias linguagens.

Uma das razões que levou a AuditMark a utilizar o PostgreSQL é o facto de ele disponibilizar ferramentas para implementar um sistema de base de dados distribuídas. Uma base de dados distribuída surge ao utilizador como se fosse uma única base de dados, mas na realidade é constituída por várias bases de dados distribuídas por diversos servidores. Algumas das vantagens do uso de base de dados distribuídas são:

- Segurança — porque temos componentes replicados, eliminando assim pontos únicos de falha;
- Aumento do desempenho — através da execução paralela (paralelismo entre consultas);
- Facilidade de expansão — aumentando-se a capacidade de processamento e armazenamento da rede;
- Custos — custa muito menos formar um sistema usando computadores menores com capacidade equivalente a uma única máquina de grande porte.

Para gerir uma base de dados distribuída, o PostgreSQL tem uma ferramenta chamada PL/Proxy. O uso do PL/Proxy trás uma grande desvantagem — desperdício de ligações, pois ele abre uma ligação por cada consulta a uma partição. Torna-se então útil, combinar o uso do PL/Proxy com o uso do *PgBouncer*, cuja função é agrupar várias consultas de várias ligações no menor número de ligações físicas às diversas partições.

3.2 Framework Javascript

Durante o desenvolvimento de uma ferramenta surge muitas vezes a necessidade de reutilizar código. Uma *framework* permite-nos reutilizar pedaços de código de uma maneira eficiente. Um dos grandes objectivos deste projecto é desenvolver uma interface interactiva e fácil de usar para isso, vamos usar muito *Javascript*. Torna-se então importante usar uma *framework* de *Javascript* que nos facilite a implementação.

Existem vários *frameworks* disponíveis, dos quais de destacam os mais populares: JQuery [19], Mootools [20] e Prototype [21]. Não é fácil escolher uma delas, pois todas disponibilizam um conjunto de recursos muito parecido. Podemos comparar o tamanho da *framework*, velocidades de

carregamento nos diferentes *browsers*, curvas de aprendizagem, complexidade das instruções, etc. No final ainda vamos achar difícil decidir-mo-nos por uma delas.

Devido a esta dificuldade em distinguir estas *frameworks*, decidi implementar um teste com cada uma delas para assim poder analisar a complexidade das instruções e conjunto de funcionalidades oferecidas por cada uma delas. Após este teste a JQuery [22] foi eleita como a melhor escolha. Na minha opinião esta é a *framework* com uma curva de aprendizagem menor e garante compatibilidade com os *browsers* dominantes. Outra das razões que me levou a optar por JQuery foi a existência de uma funcionalidade dedicada à criação de tabelas — o JQGrid.

JQGrid é uma *mini-framework*, que tem como base a JQuery, que nos permite manipular tabelas com uma enorme facilidade. Permite preencher os dados da tabelas a partir de *XML* ou *JSON*, ordenar os dados pelos diversos campos, paginar os resultados, editar e pesquisar a informação contida na tabela. Toda esta interactividade é garantida através de *AJAX*, desta forma podemos facilmente manipular e actualizar a informação contida na tabela, sem ter de recarregar a página toda.

3.3 Sistema de Renderização de Gráficos

Um dos objectivos deste projecto é obter uma interface que disponibilize a informação, tanto em forma tabular, como em forma de gráficos. Foi por isso necessário analisar as ferramentas disponíveis para a criação de gráficos em páginas *Web* a partir de informação dinâmica. As soluções encontradas foram as seguintes: AnyChart [23], Flash Charts Pro [24], amCharts [25], Fusion Charts [26], Emprise Java Script Charts [27], Swiff Chart [28] e XML/SWF [29]. Todos eles foram analisados tendo em conta os seguintes requisitos:

- Compatibilidade com os *browsers* dominantes;
- Modo de renderização;
- Modo de importação de dados;
- Controlo por *Javascript*;
- Licenciamento;
- Compatibilidade com OpenX.

Nas subsecções seguintes vamos analisar cada um destes sistemas.

3.3.1 AnyChart

Oferece compatibilidade com os *browsers* dominantes e suporte para várias linguagens como: PHP, ASP, ColdFusion ou Perl. A renderização é feita em Flash e os dados podem ser importados a partir de XML, CSV ou Javascript. Permite controlar os gráficos usando *Javascript*, é compatível com a implementação do OpenX mas não é uma ferramenta *open source*. O preço das licenças fica em \$4,999.

3.3.2 Flash Charts Pro

Oferece compatibilidade com os *browsers* dominantes e suporte para várias linguagens como: PHP, ASP, CGI ou Perl. A renderização é feita em Flash e os dados podem ser importados a partir de TXT ou XML. Permite controlar os gráficos usando *Javascript*, é compatível com a implementação do OpenX mas não é uma ferramenta *open source*. O código é disponibilizado com as licenças enterprise e worldwide. O preço das licenças fica em \$799 a Enterprise e \$1299 a Worldwide.

3.3.3 amCharts

Oferece compatibilidade com os *browsers* dominantes e suporte para várias linguagens como: PHP, .NET, JAVA ou PERL. A renderização é feita em Flash e os dados podem ser importados a partir de CSV ou XML. Permite controlar os gráficos usando *Javascript*, é compatível com a implementação do OpenX mas não é uma ferramenta *open source*. O código é disponibilizado com a licença Developer. O preço da licença é \$2750.

3.3.4 Fusion Charts

Oferece compatibilidade com os *browsers* dominantes e suporte para várias linguagens como: PHP ou JAVA. A renderização é feita em Flash e os dados podem ser importados a partir de XML. Permite controlar os gráficos usando *Javascript*, é compatível com a implementação do OpenX mas não é uma ferramenta *open source*. O preço da licença é \$1,999 mas também disponibilizam também uma versão gratuita completamente funcional mas com um menor número de gráficos disponíveis.

3.3.5 Emprise Java Script Charts

Oferece compatibilidade com os *browsers* dominantes. A renderização é feita em Flash e os dados podem ser importados a partir de XML. Permite controlar os gráficos usando *Javascript*, é compatível com a implementação do OpenX mas não é uma ferramenta *open source*. O preço da licença é \$1,000.

3.3.6 Swiff Chart

Oferece compatibilidade com os *browsers* dominantes e suporte para várias linguagens como: ASP.NET, C#, ASP.NET – VB, ASP, PHP ou JSP. A renderização é feita em Flash, png, jpeg, svg ou pdf e os dados podem ser importados a partir de um vector. A geração dos gráficos é feita do lado do servidor o que retira a possibilidade de os controlar usando *Javascript*. É compatível com a implementação do OpenX mas não é uma ferramenta *open source*. O preço da licença é \$900.

3.3.7 XML/SWF

Oferece compatibilidade com os *browsers* dominantes e suporte para várias linguagens de programação. A renderização é feita em Flash e os dados podem ser importados a partir de XML. Permite controlar os gráficos usando *Javascript*, é compatível com a implementação do OpenX mas não é uma ferramenta *open source*. O preço da licença é \$599.

3.3.8 Conclusão

Após uma análise detalhada chegou-se à conclusão que o AnyChart é demasiado caro para as funcionalidades que disponibiliza, isto porque tem de se comprar uma licença para cada servidor. O Flash Charts Pro requer a compra de uma licença especial para distribuição, o que o torna também caro mas tem a vantagem de disponibilizar o código fonte. O amCharts apresenta menos funcionalidade do que o Fusion Charts e a diferença de preços é pequena.

O Emprise Java Script Charts funciona todo ele em *Javascript*, isso levanta algumas preocupações de performance ao processar grandes quantidades de dados. Todas as demonstrações online são feitas com pequenas quantidades de dados. Além disso o Javascript muda muito de *browser* para *browser* e podemos ter problemas de compatibilidade. O XML/SWF não disponibiliza o código fonte.

De todos estes sistemas aquele que apresenta melhor relação funcionalidades/preço é o Fusion Charts. Outra vantagem deste sistema de gráficos é ter uma versão gratuita com as funcionalidades básicas necessárias para a realização deste projecto. Apresenta também uma óptima documentação e a possibilidade de *upgrade* a qualquer altura sem necessidade de grandes alterações.

Ficou assim decidido que o sistema de gráficos a utilizar será o Fusion Charts, versão gratuita.

3.4 Framework de apoio ao desenvolvimento

Todas as três camadas do modelo *MVC* têm de ser integradas. Pode-se fazer isso “à mão” ou utilizando uma ferramenta chamada *framework*. Não existe uma definição oficial de *framework* no entanto pode-se citar uma definição dada por Fayad e Schmidt [30] que é a seguinte:

“Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação.”

Na realidade um *framework* permite implementar o modelo *MVC* de uma maneira simples e quase transparente para o programador. É por isso utilizado em grande parte do desenvolvimento para a *Web*.

Existem actualmente dezenas de *frameworks* diferentes, cada um deles desenvolvido e optimizado para trabalhar com uma linguagem de programação *web* diferente. Existem também diversas linguagens de programação *web* tais como:

- PHP

- ASP
- .NET
- Ruby
- JSP

Para este projecto vai usar-se *PHP* [31] como linguagem de programação. O *PHP* foi escolhido pela sua facilidade de utilização, performance e experiência pessoal de todos os envolvidos no projecto. Para a linguagem *PHP* existem ainda assim diferentes *frameworks* disponíveis. Foi então necessário fazer um estudo das características de cada um de modo a que se pudesse escolher o que melhor responde às necessidades do projecto. Devido à grande quantidade de soluções disponíveis a análise foi feita em duas fases. Numa primeira fase foram considerados os diferentes *frameworks* disponíveis para *PHP*:

- Akelos [32]
- ash.MVC [33]
- CakePHP [34]
- CodeIgniter [35]
- DIY [36]
- eZ Components [37]
- Fusebox [38]
- PHP on TRAX [39]
- PHPDevShell [40]
- PhpOpenbiz [41]
- Prado [42]
- QPHP [43]
- Seagull [44]
- Symfony Project [45]
- WACT [46]
- WASP [47]
- Yii [48]

- Zend [49]
- ZooP [50]

Nesta primeira fase de comparação tentou-se descobrir quais os *frameworks* que:

- São compatíveis com *PHP4*
- São compatíveis com *PHP5*
- Suportam o modelo *MVC*
- Suportam diferentes tipos de base de dados sem necessário fazer alterações
- Suporta ORM²
- Têm DB Objects³
- Têm um *template engine* incorporado
- Têm alguma maneira de fazer *cache* de páginas
- Têm um componente de validação de formulários
- Têm AJAX incorporado
- Têm um módulo de autenticação de utilizadores
- Têm outros módulos como *RSS feed parser* ou *PDFModules*
- Têm EDP⁴

Os resultados desta fase de análise podem ser consultados na Tabela 3.1 e na Tabela 3.2.

Após a primeira fase de comparação escolheram-se 4 *frameworks* que se destacaram pelas suas características:

1. CakePHP
2. CodeIgniter
3. Symfony Project
4. Zend

²*Object-record mapper* — normalmente uma implementação de ActiveRecord

³outros objectos de base de dados como TableGateWay

⁴Event Driven Programming — esta técnica é muito recente

Tabela 3.1: Primeira fase de comparação de frameworks

Framework	PHP4	PHP5	MVC	DB's	ORM	DB Object
Akelos	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
ash.MVC	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>SIM</i>
CakePHP	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
CodeIgniter	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>
DIY	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>
eZ Components	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>
Fusebox	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>
PHP on TRAX	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
PHPDevShell	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>NÃO</i>	<i>NÃO</i>
PhpOpenbiz	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
Prado	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
QPHP	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>
Seagull	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
Symfony Project	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
WACT	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>
WASP	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>
Yii	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
Zend	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
ZooP	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>

Pode ver-se nas tabelas que o *CakePHP* possui todas as características consideradas excepto o *template engine*, mas este pode ser integrado com facilidade. O mesmo acontece com o *Symfony*. O *CodeIgniter* e o *Zend* suportam a maioria das funcionalidades e aquelas que não vêm incorporadas são facilmente integradas; o mesmo não acontece com o resto dos *frameworks*. Pode ver-se também que apenas 3 dos frameworks vêm com *EDP*, que como já foi referido é uma tecnologia recente. Alguns dos *frameworks* foram excluídos, não só baseado nas características descritas nas tabelas mas devido a outras características como desempenho ou documentação, nas quais estes 4 se destacam.

Para podermos escolher o melhor destes últimos 4 *frameworks* foi necessário analisar as seguintes características:

- Search — Tem ou não métodos integrados que facilitam a pesquisa por palavras-chave dentro do *site*;
- Webservices — Tem ou não serviços *Web* incorporados;
- Multilíngua — Tem ou não métodos que facilitam a implementação do *site* em diferentes línguas;
- Debug — Tem ou não ferramentas para fazer *debug*;
- MAIL — Tem ou não métodos que facilitam o envio de email utilizando o protocolo **SMTP**⁵.

⁵Simple Mail Transfer Protocol

Tabela 3.2: Primeira fase de comparação de frameworks(Cont.)

Framework	Templates	Caching	Validation	AJAX	Auth Module	Modules	EDP
Akelos	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
ash.MVC	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
CakePHP	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
CodeIgniter	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>NÃO</i>	<i>NÃO</i>
DIY	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>NÃO</i>
eZ Components	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>NÃO</i>	<i>NÃO</i>
Fusebox	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>
PHP on TRAX	<i>NÃO</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>
PHPDevShell	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
PhpOpenbiz	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>
Prado	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
QPHP	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
Seagull	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
Symfony Project	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
WACT	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>SIM</i>	<i>NÃO</i>
WASP	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
Yii	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
Zend	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
ZooP	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>

Os resultados desta segunda fase de comparação podem ser consultados na tabela 3.3. Pode-se ver que o *CakePHP* não possui serviços de *MAIL*, o *CodeIgniter* não tem *webservices* integrados nem suporte para multilíngua e o *Symfony* não têm integrado um módulo que permite efectuar pesquisa nas páginas. Desta forma, concluiu-se que a melhor escolha seria o *ZEND*, pois foi o único que satisfaz todos os requisitos.

Tabela 3.3: Segunda fase de comparação de frameworks

Framework	Templates	Caching	Validation	AJAX	Auth Module
CakePHP	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>NÃO</i>
CodeIgniter	<i>SIM</i>	<i>NÃO</i>	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>
Symfony Project	<i>NÃO</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>
Zend	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>	<i>SIM</i>

3.5 Testes de frameworks

Quando se escolhe um *framework* não se pode ter em conta apenas as funcionalidades disponibilizadas, também se deve analisar a complexidade e a curva de aprendizagem que cada *framework* exige. Para melhor fundamentar a minha escolha decidi implementar um pequeno teste com cada *framework*.

O teste consistiu em criar uma tabela na base de dados e criar um conjunto de páginas que permitiam inserir, pesquisar e apagar informação da base de dados.

Após a conclusão destes testes verificou-se que é fácil colocar qualquer um deles a funcionar correctamente. Porém tive algumas dificuldades em executar o teste com o *CodeIgniter*. O modo de funcionamento do *Symfony* e do *CakePHP* é bastante intuitivo e aprende-se bastante depressa. O modo de funcionamento do *Zend* é mais complicado, para alguém que está a trabalhar com ele pela primeira vez. Esta complexidade deve-se ao facto de ser um *framework* mais completo e por isso ter uma curva de aprendizagem maior.

Em suma, após o teste de implementação, a melhor escolha continua a ser o *Zend* [51].

Capítulo 4

Especificação de Requisitos

Este capítulo descreve os requisitos funcionais e não funcionais deste projecto. Devido à complexidade do problema, decidiu-se dividir o problema em pequenas partes e analisar os requisitos de cada parte. Os requisitos do sistema foram divididos da seguinte maneira:

- Site Corporativo / Registo e login (Ver subsecção 4.1)
- Acessos e permissões (Ver subsecção 4.2)
- Interface de análise (Ver subsecção 4.3)
- Documentação e suporte (Ver subsecção 4.4)

Nas secções seguintes vai descrever-se cada um destes módulos, analisando os requisitos funcionais e não funcionais. Para melhor ilustrar o problema vamos também apresentar diagramas de casos de utilização.

4.1 Site Corporativo / Registo e login

Pretende-se construir um sítio *Web* tendo como ponto de partida o sítio *Web* actual da empresa, no qual se vai disponibilizar alguma informação sobre o funcionamento da empresa, contactos disponíveis, alguns documentos com notícias importantes, um motor de busca para pesquisar conteúdos dentro do sítio *Web* e uma área de clientes onde os clientes podem efectuar o seu registo ou login.

Pretende-se implementar um sistema de gestão de utilizadores, que permita registo autónomo e recuperação de password por email. Este sistema deve permitir a importação dos dados pessoais a partir do OpenID e verificar todos os dados em tempo real mostrando ao utilizador os possíveis erros. O sistema de registo deve conter um *CAPTCHA*¹ para evitar a interpretação por máquinas.

¹Completely Automated Public Turing test to tell Computers and Humans Apart

A aceitação de um registo pode ser automática ou manual e só se pode fazer login após a validação do email.

4.2 Acessos e permissões

Após o registo, é criada uma conta de utilizador. Dentro desta conta de utilizador podem existir várias campanhas que podem ser agrupadas em contas de campanhas, permitindo assim associar campanhas que têm um denominador comum.

Cada campanha tem associado um grupo de permissões. Estas permissões apenas podem ser alteradas pelo dono da campanha. Cada utilizador, quando cria uma conta de campanhas, fica automaticamente como dono dessa conta e de todas as campanhas que pertencem a essa conta.

Por omissão só tem acesso à informação de uma campanha o seu dono. Este pode depois definir permissões para outros utilizadores do sistema. Estas permissões podem ser administrador ou convidado. A permissão administrador permite ao utilizador realizar acções administrativas sobre a campanha, enquanto que a permissão convidado apenas permite ao utilizador visualizar a informação da respectiva campanha.

As campanhas são identificadas a partir do seu nome e ID único. Deve ser possível importar campanhas a partir de *Ad Servers* externos como:

- Openx
- Google Adwords e Adsense
- Microsoft AdCenter
- Yahoo Ads

4.3 Interface de análise

A interface deve ter uma página inicial igual para todos os utilizadores onde aparece um resumo com a informação mais importante. A informação que se mostra na página inicial deve poder ser personalizada e deve existir um menu, sempre visível, com acesso às configurações pessoais e da conta.

Os menus devem ser simples e intuitivos, mostrando poucas opções de cada vez para não confundir o utilizador e devem ter *links* de acesso rápido às várias funcionalidades. Deve também existir um *dashboard* onde se mostra os diferentes tipos de análises disponíveis. Este *dashboard* deve ser personalizável.

As análises podem ser feitas individualmente por campanha ou o utilizador pode seleccionar/agregar várias campanhas e efectuar análises comparativas ou somatório dessas campanhas. Os gráficos devem ser interactivos sendo possível escolher o intervalo de tempo que queremos analisar (mês, semana, dia ou hora) e deve ser possível comparar gráficos de campanhas diferentes e/ou intervalos de tempo diferentes.

Todas as análises devem ser disponibilizadas em forma de tabelas ou em diferentes tipos de gráficos (circular, barras, linhas) com a possibilidade de exportar os dados para *XLS*, *CVS*, *XML* ou *PDF*. Também deve ser possível ordenar a informação pelos diferentes campos e enviar dados por email, podendo agendar o envio automático (por exemplo todas as semanas).

As análises devem fornecer informação sobre número de visitantes, bem como toda a informação do visitante (país de origem, *browser* utilizado, tendências, velocidade da ligação, etc), origens do tráfego (sites referência, motores de busca, etc...), *IP* ou *subnet*, etc.

Toda a interface deve ser multilíngua (pelo menos Português e Inglês) e deve existir um menu de ajuda contextual, que muda a sua informação de acordo com a funcionalidade que está a ser utilizada.

4.4 Documentação e suporte

Pretende-se também integrar um sistema de Ajuda e FAQ², que deve permitir pesquisar e avaliar cada artigo. Devem existir *links* para o sistema de Ajuda em todas as páginas e o utilizador deve ter a hipótese contactar a empresa por email ou telefone para esclarecer dúvidas ou deixar uma sugestão.

Também faz parte dos requisitos disponibilizar pequenos tutoriais de demonstração (possivelmente em vídeo) das ferramentas, para uma fácil adaptação dos utilizadores.

4.5 Casos de Utilização

A melhor maneira de descrever os requisitos de um sistema é a partir de diagramas *UML* [52] de casos de utilização. Para melhor ilustrar o problema decidimos efectuar um conjunto de diagramas. Devido à grande quantidade de casos de utilização, para tornar os diagramas mais legíveis decidiu-se agrupar os casos de utilização em pacotes. Os pacotes considerados foram: Site Corporativo, Área de Cliente, Interface de Análise, Interface de Administração e Sistema de Ajuda como se pode ver nas figuras seguintes.

²Frequently Asked Questions

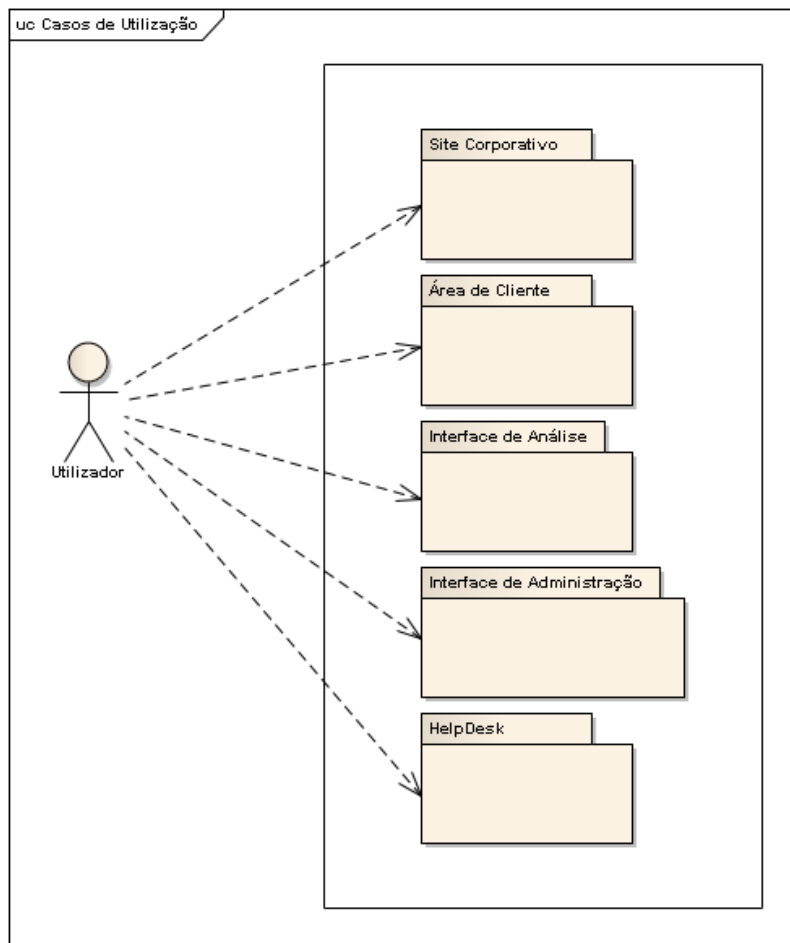


Figura 4.1: Diagrama de Pacotes

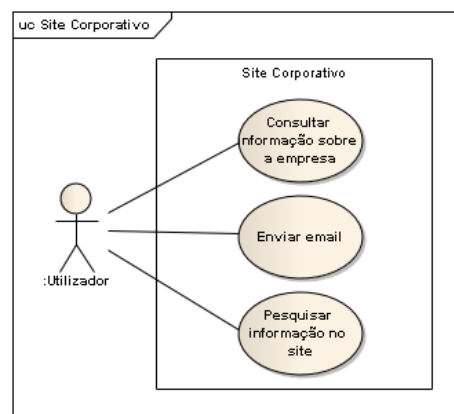


Figura 4.2: Casos de utilização — Site Corporativo

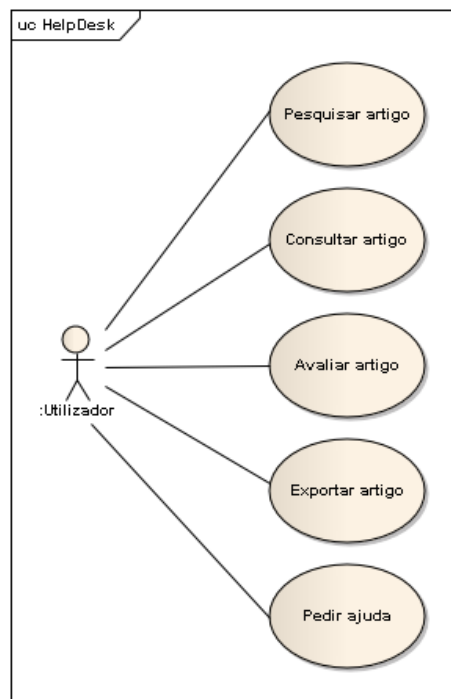


Figura 4.3: Casos de utilização — Sistema de Ajuda

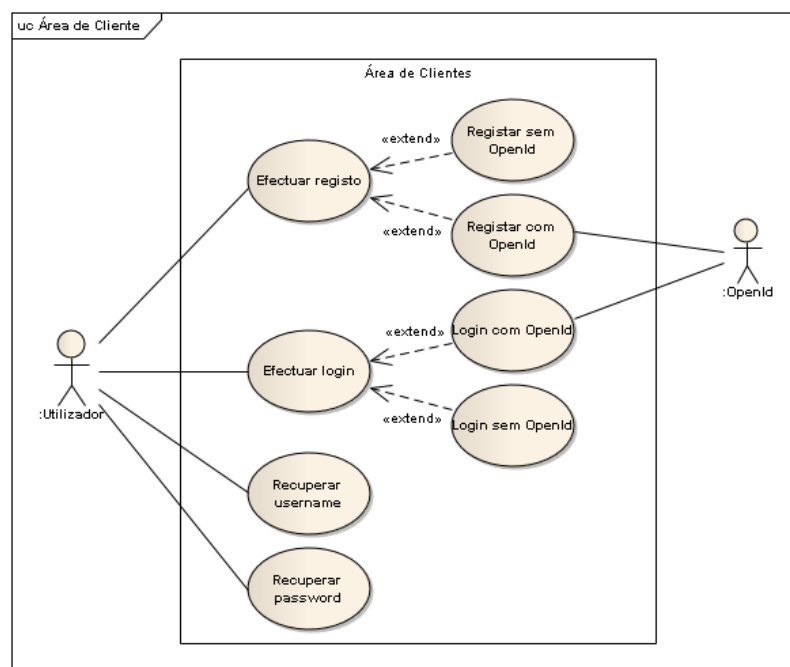


Figura 4.4: Casos de utilização — Área de Cliente

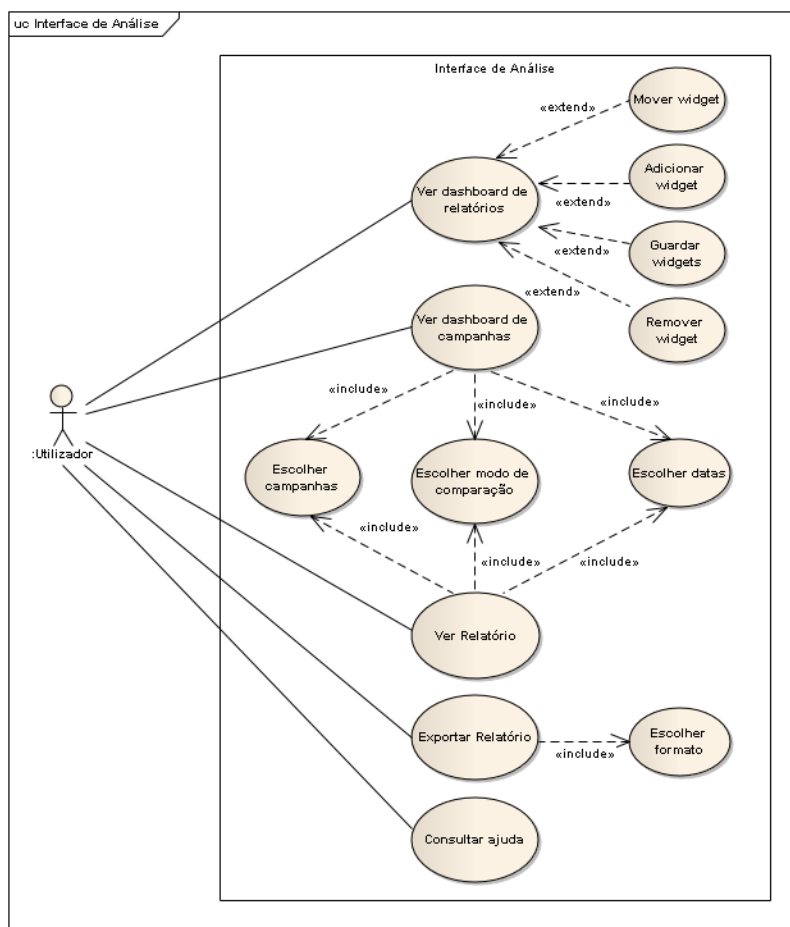


Figura 4.5: Casos de utilização — Interface de Análise

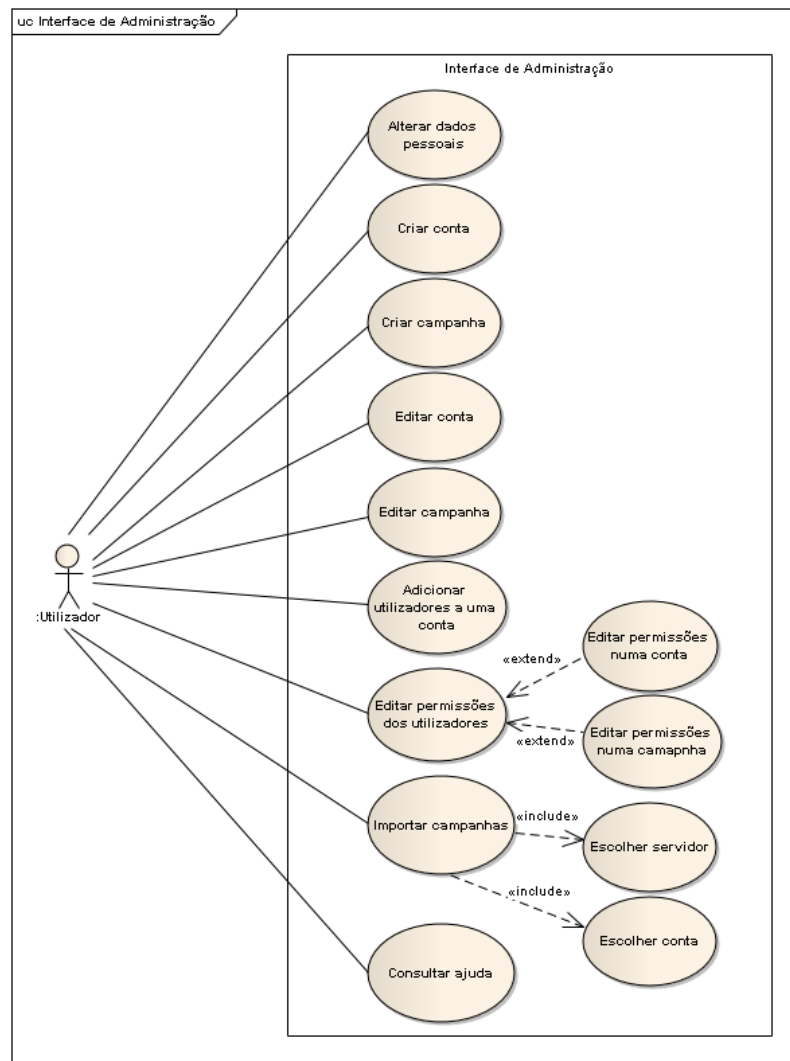


Figura 4.6: Casos de utilização — Interface de Administração

Capítulo 5

Implementação

Este capítulo descreve a solução encontrada e os detalhes da implementação deste projecto.

5.1 Arquitectura do Sistema

Este sistema é constituído por uma interface *Web* desenvolvida em *HTML*, *CSS* e *Javascript*, uma camada de lógica de negócio desenvolvida em *PHP* e uma base de dados *PostgreSQL*.

5.1.1 Arquitectura Física

Para hospedar a interface foi utilizada uma máquina com sistema operativo *Linux/Ubuntu* com um servidor *Apache*. Esta máquina vai ser acedida a partir da *Internet* e vai ter acesso a uma rede privada onde vão estar o resto das máquinas que contêm as base de dados (ver Figura 5.1).

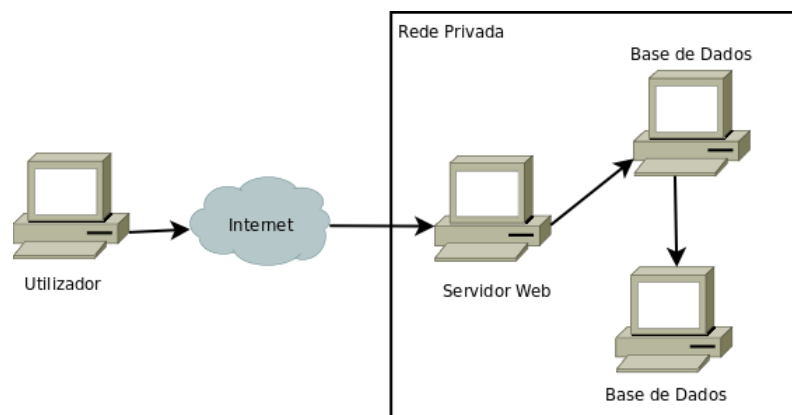


Figura 5.1: Arquitectura Física

Neste projecto vamos utilizar uma lógica de bases de dados distribuídas. Para isso vamos criar duas base de dados diferentes — plproxy e auditservice. A plproxy vai gerir o acesso às diferentes partições que vão ser criadas, enquanto que a auditservice vai guardar a informação.

Após analisar a informação que foi criada para testar a aplicação chegou-se à conclusão que, devido à grande quantidade de informação que existia sobre a análise de campanhas quando comparada com o resto da informação, apenas a esta deveria ficar segmentada. Todo o resto da informação vai ficar replicada nas diversas máquinas, para garantir a coerência do sistema. Existem 2 tipos de segmentação: horizontal, dividindo as tabelas por tuplos, e vertical, dividindo as tabelas por colunas. Devido à forte dependência temporal da informação relativa às campanhas, decidiu-se implementar a seguinte lógica de segmentação horizontal: uma máquina vai guardar a informação relativa aos dias pares, enquanto que a outra vai guardar a informação relativa aos dias ímpares. Desta forma quando chegar um pedido, ele vai ser difundido pelas duas máquinas que constituem o *cluster*, ou seja, o nosso pedido vai ser processado por cada uma das máquinas em paralelo, diminuindo o tempo dos pedidos e aumentando o desempenho do sistema. Do ponto de vista do utilizador, é como se existisse apenas uma base de dados, pois o PL/Proxy cria uma camada de abstracção de toda a base de dados.

Todos os acessos à base de dados são feitos através de um conjunto de *procedures*¹ disponibilizadas pela base de dados plproxy. O esquema de acesso é o seguinte: a camada de dados da aplicação faz um pedido à base de dados plproxy chamando uma das *procedures* disponíveis; este pedido é depois feito em paralelo às duas base de dados auditservice; cada uma delas responde com a informação pedida; esta informação é agrupada e ordenada na plproxy e depois devolvida à camada de dados da aplicação.

Toda a comunicação entre o cliente e o servidor web pode ou não ser feita usando encriptação. O sistema está preparado para trabalhar com ou sem SSL, sendo uma escolha do utilizador se vai utilizar ou não. Na comunicação entre o servidor web e a base de dados não houve preocupações de segurança, pois assumimos que ambas as máquinas vão estar dentro de uma rede privada.

5.1.2 Arquitectura Lógica

Como foi referido nos capítulos anteriores, a aplicação vai ser implementada usando o modelo MVC. As camadas *Controller* e *Model* vão ser desenvolvidas em PHP enquanto que a camada *View* vai ser desenvolvida em *HTML*, *CSS* e *Javascript*. Devido à complexidade da aplicação decidiu-se dividir a camada *Controller* em vários módulos: sítio *Web*, área de cliente, interface de análise, interface de administração e sistema de ajuda. Cada um destes módulos é constituído por várias classes, controladores, e cada classe tem vários métodos — acções. Uma acção corresponde a um pedido, do qual pode ou não resultar página *Web*. Um controlador corresponde a um conjunto de acções que estão relacionadas. O critério de agregação de acções foi a sua funcionalidade. A camada *Model* contém as classes que disponibilizam o acesso à base de dados. Esta camada garante a separação entre os dados e a sua visualização.

¹Funções, métodos

5.2 Modelo de Dados

O modelo de dados é uma ferramenta de modelação importante porque os seus conceitos básicos são simples e gerais e porque o seu desenho não depende de nenhum tipo de programa informático. O modelo de dados apresenta os dados como um conjunto de relações. Para representar o sistema foi criado o modelos de dados apresentado na Figura 5.2.

Um cliente é definido por um *id* único, um *username* único, uma *password*, nome completo, número de contribuinte, morada, cidade, país, código postal, língua, telefone, fuso horário, moeda, *openid* se for o caso, saldo da conta, *accessKey* e *secretKey* para poder aceder à sua conta através de *web services*. Esta informação é guardada na tabela *client*.

Quando um cliente acede à aplicação é criada uma sessão que fica ligada ao *browser* e *ip* utilizados pelo utilizador. O *id* de sessão é depois utilizado para identificar os pedidos feitos pela interface. Esta informação fica guardada na tabela *session*. Cada cliente pode configurar o *dashboard* ao seu gosto. As configurações são guardadas na tabela *dashboardConfig* para estarem disponíveis da próxima vez que aceder à interface.

Uma conta representa um conjunto de campanhas que estão relacionadas e é descrita através de um *id* único, um nome, um dono e um estado como se mostra na tabela *account*. Uma conta só pode ter um dono, o utilizador que a criou, mas pode ser acedida por vários utilizadores, se o dono assim o decidir. Esta relação está definida na tabela *accountUsers*. O tipo de acesso pode ser *guest* ou *administrator*.

Uma campanha é definida pelo seu *id* único, nome, descrição, status, nome do servidor onde foi criada, *id* da campanha no servidor em que foi criada, *id* da conta a que pertence, data de início e data final da campanha. Esta informação foi modelada na tabela *campaign*. Cada campanha só pode pertencer a uma conta. O tipo de acesso de um utilizador a uma conta e respectivas campanhas pode ser definido globalmente para todas as campanhas de uma conta ou pode ser definido um tipo de acesso específico para cada campanha utilizando a tabela *permission*.

Uma factura é representada pelo seu cabeçalho e pelo seu corpo. O cabeçalho contém um *id* único, o *id* do cliente, intervalo de datas a que diz respeito e um total. O corpo da factura é constituído por vários itens. Cada item tem um *id* do item, uma descrição do item, a taxa aplicada a esse item, o preço unitário e a quantidade. Esta informação foi modelada em três tabelas: *billHeader*, *billbody* e *billElements*.

A aplicação vai disponibilizar 6 tipos de análises diferentes. Os resultados de cada análise são guardados nas seguintes tabelas: *repBrowser*, *repSysOp*, *repLanguage*, *repCountry*, *repFraudType* e *repClicksTime*. As diferentes possibilidades para cada um dos relatórios são guardadas em tabelas separadas que depois são referenciadas através de um *id* único.

5.3 Esboço da Interface Gráfica

Grande parte do esforço do desenvolvimento dedicado a este projecto foi colocado na sua interface gráfica. O objectivo era desenvolver uma interface intuitiva, simples de usar, dinâmica e



Figura 5.3: Esboço do Site Corporativo

com um custo de aprendizagem pequeno. Para analisar e corrigir possíveis problemas, decidiu-se efectuar um esboço antes de começar a implementar. Isto é útil pois assim ficamos com uma ideia de como a interface vai ficar e no caso de ser necessário efectuar alterações, o custo é muito menor do que se for feito depois de estar implementada.

Existem muitas ferramentas disponíveis para efectuar prototipagem de interfaces gráficas, desde o Powerpoint até soluções especializadas, todas elas permitem criar um protótipo funcional em pouco tempo. A maior diferença entre elas é o conjunto de ferramentas disponibilizadas. A nossa escolha recaiu sobre o Serena [53]. É um software grátis que oferece um vasto conjunto de funcionalidades dedicadas à prototipagem de interfaces web tais como botões, hiperligações, navegação entre páginas, possibilidade de criar e simular cenários condicionados por variáveis definidas pelo utilizador em runtime, etc. Nas Figuras 5.3, 5.4, 5.5 e 5.6 são apresentados alguns *screenshots*² do protótipo criado.

5.4 Módulos

5.4.1 Sítio Web

O ponto de entrada para a interface da aplicação é o sítio Web da empresa. A primeira tarefa foi redesenhar o sítio Web da empresa para ficar a condizer com o modelo gráfico escolhido para a interface. Este sítio web disponibiliza aos utilizadores alguma informação geral sobre a empresa,

²Capturas do ecrã

Settings - Personal Info

AUDITMARK

username | Home | Administration | Help | Sign Out

Administration -> Personal

Personal

General info

Change Password

Account

Campaigns

Billing

Help

About this action

Common questions

Ask for support

Email: baldasman@gmail.com

Username: baldasman

Full Name: Pedro Santos

Phone: 123456789

VAT: 1234567890

Address: Rua qqcoisa

Country: USA

State: Florida

City: Miami

Zip Code: 12345

Currency: Dollar

Time Zone: Pacific

Language: English

Save

© 2009 Auditmark

Figura 5.4: Esboço da Interface de Administração

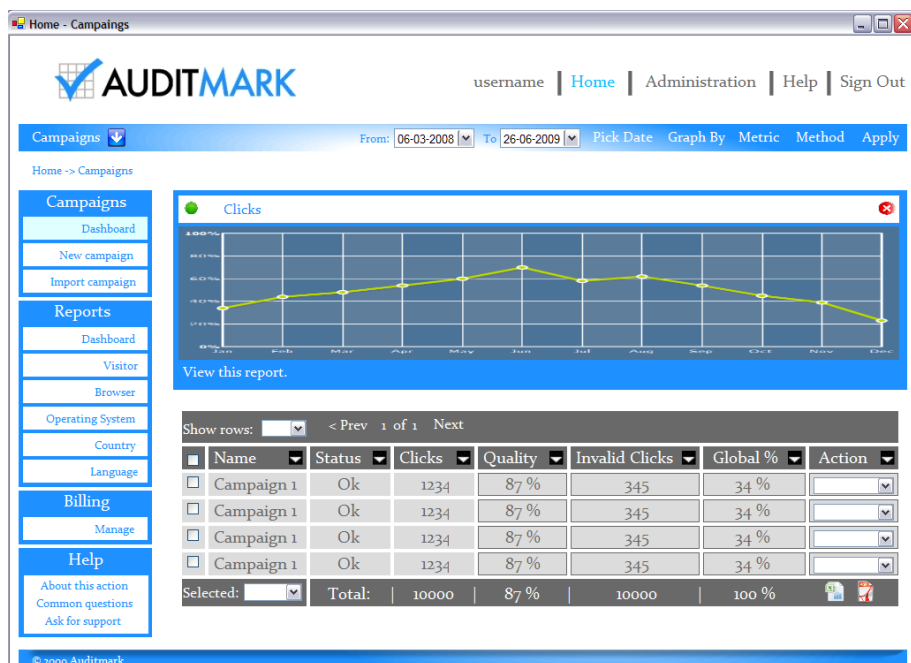


Figura 5.5: Esboço do Dashboard de Campanhas

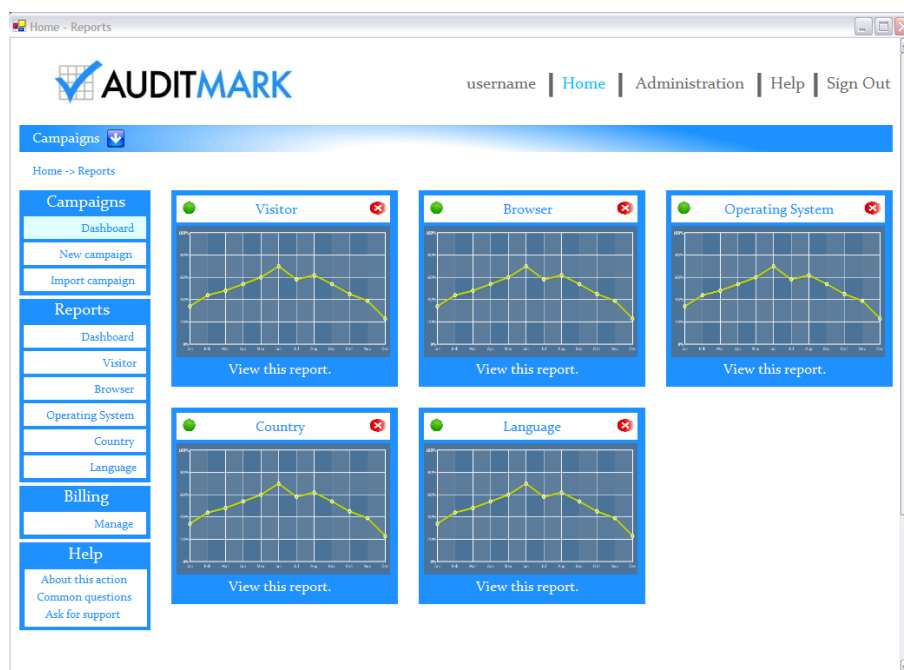


Figura 5.6: Esboço do Dashboard de Relatórios

contactos disponíveis, dá a possibilidade ao utilizador de enviar questões por email à empresa e disponibiliza também um motor de busca de conteúdos dentro do sítio *Web*. O sistema de envio de email foi cuidadosamente pensado para evitar a sua interpretação por máquinas, evitando assim email indesejados. Para isso recorreu-se a técnicas de disfarce de email utilizando *Javascript* para decodificar os endereços de email. Os formulários também foram desenvolvidos recorrendo a *Javascript* para permitir dar um *feedback* ao utilizador de possíveis erros no formulário antes da sua submissão.

O motor de busca de conteúdos foi implementado utilizando uma funcionalidade disponibilizada pela framework Zend. Esta funcionalidade utiliza o motor de busca Lucene que consegue gerar índices de conteúdos a partir de páginas *Web*. O modo de funcionamento do Lucene levou a outra necessidade — criar um *sitemap* para alimentar a criação dos índices de pesquisa.

Para criar o *sitemap* dinamicamente foi desenvolvido um *script* em Perl que usa o comando *wget* para descobrir as diversas páginas disponíveis num sítio web. Para tornar este procedimento completamente transparente tirou-se partido do *cronjob* disponível em Linux para agendar tarefas. Assim sendo, sempre que se adicionar novas páginas ao sítio *Web*, o programador não precisa de se preocupar com os índices de pesquisa, pois estes vão ser reconstituídos automaticamente.

Figura 5.7: Formulário de Registo

5.4.2 Área de Clientes

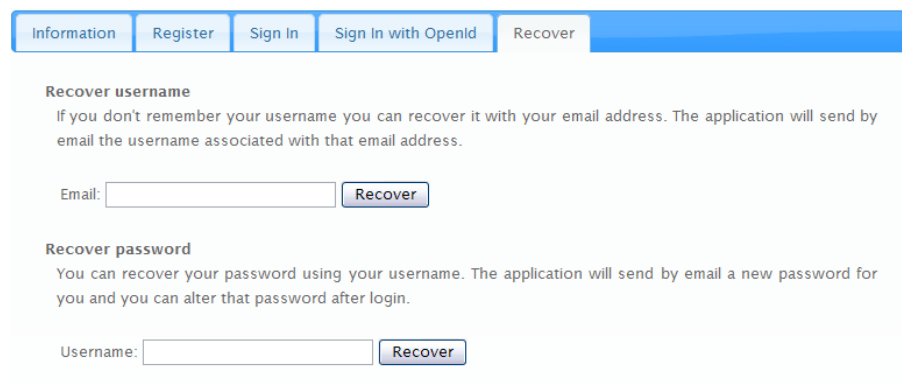
5.4.2.1 Autenticação

O primeiro grande desafio foi desenvolver o sistema de autenticação. O objectivo era desenvolver um sistema fiável baseado em sessões para evitar o constante envio do *username* e *password* do utilizador. Quando o utilizador efectua o login, é calculada uma hash (utilizando o algoritmo MD5) do seu *username* + *password* e é enviado para o servidor o seu *username* e o hash calculado. No lado da base de dados é calculado o hash do *username* e *password* que estão guardadas e é depois comparado com o que foi recebido. Se a comparação for bem sucedida, é criada uma nova sessão à qual é atribuído um *id* único, gerado aleatoriamente, com 32 caracteres. Este *id* é depois devolvido e guardado numa variável de sessão do PHP. Sempre que um utilizador fizer um pedido, apenas é enviado o *id* de sessão juntamente com um hash do seu *IP* (para garantir que a sessão não é roubada) e nunca o seu *username* e *password*.

5.4.2.2 Registo e Recuperação de Username/Password

Desenhar formulários simples e fáceis de preencher é um grande desafio. Na implementação do formulário de registo (ver Figura 5.7) foi integrado um sistema de ajuda contextual para ajudar os utilizadores a preencher facilmente e sem equívocos a informação pedida. Este sistema é capaz de identificar qual o campo do formulário que está a ser preenchido e fornece pequenas dicas para o seu correcto preenchimento. É feita uma verificação de todos os campos à medida que o utilizador vai preenchendo a informação. Isto foi conseguido usando *Javascript* nas verificações que necessitam de aceder à base de dados, como é o caso do email ou *username*. Também foi incluído um sistema de *CAPTCHA*, para evitar a interpretação por máquinas.

O sistema de recuperação de dados (ver Figura 5.8) foi partido em dois devido a questões de segurança. O utilizador pode recuperar a sua *password* a partir do *username* ou o *username* a partir do seu email. Em ambos os casos a informação é enviada para o email do utilizador. No



The image shows a web interface with a navigation bar at the top containing five tabs: 'Information', 'Register', 'Sign In', 'Sign In with OpenId', and 'Recover'. The 'Recover' tab is currently selected and highlighted in blue. Below the navigation bar, there are two sections for recovery:

- Recover username**: A text block stating, 'If you don't remember your username you can recover it with your email address. The application will send by email the username associated with that email address.' Below this text is a form with an 'Email:' label, an input field, and a 'Recover' button.
- Recover password**: A text block stating, 'You can recover your password using your username. The application will send by email a new password for you and you can alter that password after login.' Below this text is a form with a 'Username:' label, an input field, and a 'Recover' button.

Figura 5.8: Formulário de Recuperação Username/Password

caso da recuperação da *password*, como o próprio sistema não conhece a *password* do utilizador, é gerada uma nova *password* que é enviada para o utilizador. Este pode depois alterá-la na interface de administração.

5.4.2.3 Integração com OpenId

Fazia parte dos objectivos permitir aos utilizadores registarem-se e fazerem login usando o seu OpenId. Esta funcionalidade foi implementada usando uma ferramenta disponibilizada pelo Zend. O utilizador introduz o seu openid e a interface envia um pedido de autenticação para o servidor responsável pela identidade. O utilizador deve depois autenticar-se no servidor do openid. Após uma autenticação bem sucedida o utilizador é redireccionado para a interface que analisa a resposta e em caso de sucesso verifica se o openid existe na sua base de dados. No caso do registo, a interface ao enviar o pedido de autenticação envia também um pedido adicional de informação do utilizador. Se o utilizador autorizar o envio da informação pedida o servidor do openid devolve essa informação à interface que preenche o formulário de registo automaticamente. Resta ao utilizador preencher a informação em falta e submeter o formulário. Desta forma o utilizador não precisa de criar uma *password* para a interface, pois a sua autenticação é feita pelo servidor do openid.

5.4.3 Interface de Administração

Esta área da aplicação permite aos utilizadores:

- Editar os seus dados pessoais;
- Criar e editar contas;
- Criar, editar e importar campanhas;
- Partilhar informação com outros utilizadores;

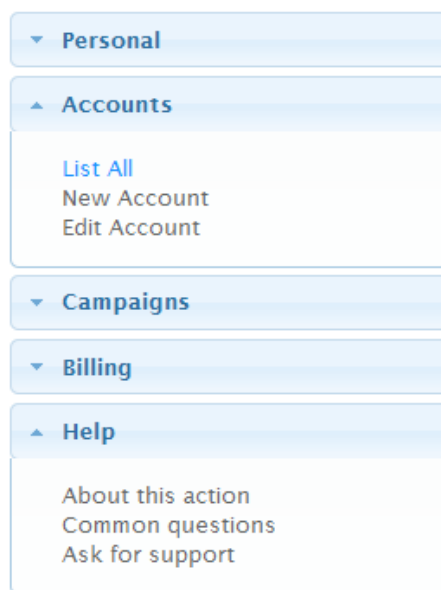


Figura 5.9: Menu Lateral da Interface de Administração

- Visualizar facturas.

Para assegurar uma fácil navegação ao longo de todas as funcionalidades disponibilizadas neste ecrã, foi criado um menu lateral com efeito de acordeão, como se mostra na Figura 5.9. Este é um dos efeitos visuais disponibilizado pela *framework* jQuery, que permite esconder os submenus que não estão a ser utilizados. Este efeito permite colocar um menu de grande dimensão num pequeno espaço. O utilizador pode alternar a visibilidade dos submenus clicando no nome do submenu.

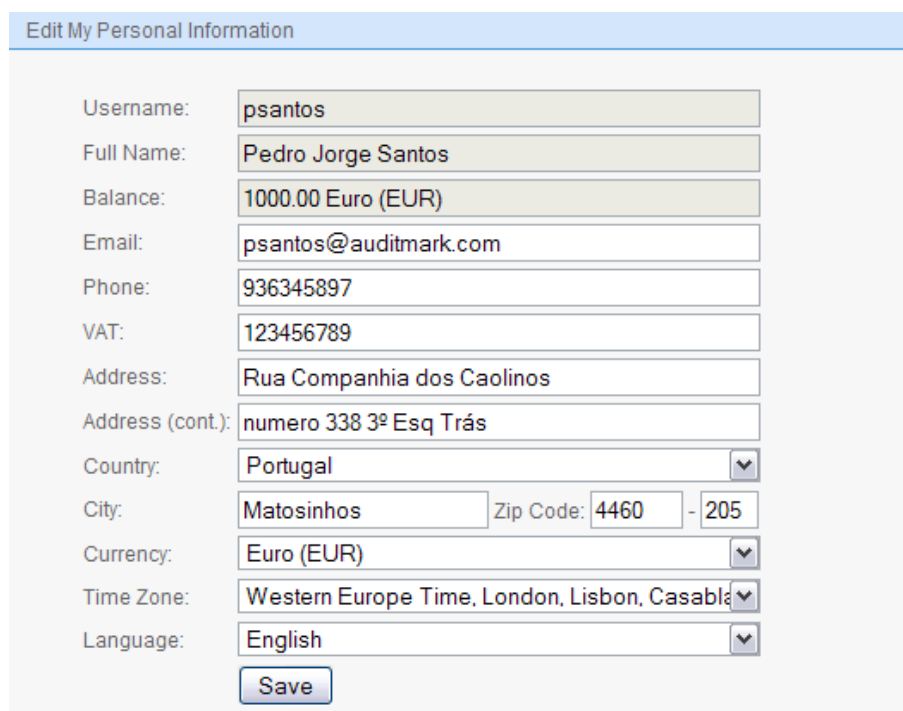
5.4.3.1 Configurações Pessoais

O utilizador pode facilmente completar ou editar as suas informações pessoais a partir de um formulário disponível como se pode ver na Figura 5.10. Cada vez que o utilizador submete este formulário o sistema verifica se toda a informação necessária está preenchida. No caso de haver informação em falta a conta do utilizador fica bloqueada e o utilizador não pode aceder a outras zonas da aplicação enquanto a situação não for regularizada.

Toda a interface é multi-língua e está preparada para disponibilizar relatórios em diferentes fusos horários. É aqui que o utilizador define as suas preferências. Todas as preferências são carregadas quando o utilizador efectua login e estão disponíveis no resto da aplicação a partir de variáveis de sessão.

5.4.3.2 Gestão de Contas e Campanhas

Cada utilizador pode ter várias contas, e cada conta pode ter muitas campanhas. A solução encontrada para oferecer uma gestão simples de todos estes dados passou pelo uso de tabelas com



The image shows a web form titled "Edit My Personal Information". It contains several input fields for personal data: Username (psantos), Full Name (Pedro Jorge Santos), Balance (1000.00 Euro (EUR)), Email (psantos@auditmark.com), Phone (936345897), VAT (123456789), Address (Rua Companhia dos Caolinos), Address (cont.) (numero 338 3º Esq Trás), Country (Portugal), City (Matosinhos), Zip Code (4460 - 205), Currency (Euro (EUR)), Time Zone (Western Europe Time, London, Lisbon, Casabl), and Language (English). A "Save" button is at the bottom.

Username:	psantos		
Full Name:	Pedro Jorge Santos		
Balance:	1000.00 Euro (EUR)		
Email:	psantos@auditmark.com		
Phone:	936345897		
VAT:	123456789		
Address:	Rua Companhia dos Caolinos		
Address (cont.):	numero 338 3º Esq Trás		
Country:	Portugal		
City:	Matosinhos	Zip Code:	4460 - 205
Currency:	Euro (EUR)		
Time Zone:	Western Europe Time, London, Lisbon, Casabl		
Language:	English		
<input type="button" value="Save"/>			

Figura 5.10: Formulário de Edição dos Dados Pessoais

drill-down como se mostra na Figura 5.11. Esta é uma das funcionalidades disponibilizada pela *framework* jqGrid, que consiste em usar várias tabelas, cada uma com um nível de detalhe maior do que a anterior.

Todas as tabelas permitem fazer paginação dos resultados, decidir quantos resultados pretendemos visualizar de cada vez e ordenar os resultados pelos diversos campos da tabela. Algumas das tabelas, permitem também adicionar, editar e apagar resultados através de pequenos formulários interactivos como se mostra na Figura 5.12.

5.4.3.3 Importação de Campanhas

De modo a permitir a importação de campanhas a partir de servidores externos foi necessário fazer um levantamento do modelo de dados dos diversos servidores (ver Anexo A). Devido a limitações de tempo, ficou decidido que só vai ser possível importar campanhas a partir do Google Adwords e do OpenX.

Existem duas alternativas para a importação de campanhas, pode ser feita através de *Web Services* ou a partir de um relatório. A utilização de *Web Services* foi descartada porque exigia que os utilizadores divulgassem as suas credencias de acesso ao servidor, dando total acesso às suas contas e informações, o que levantava problemas de segurança e privacidade. Foi então implementado um sistema de importação baseado em relatórios. Estes relatórios podem ser obtidos, pelos utilizadores, junto do seu servidor sem por em causa toda a sua informação, pois no relatório apenas aparece a informação necessária à importação.

My Accounts

List of Accounts

Account Name	Status	Number of Campaigns	Action
default	Running	0	<div></div>
Publicidade Automóvel 3	Running	1	<div></div>

1

of 1

10

2 Records

List Of Campaigns For default

Campaign Name	Status	Action
Publicidade TV	Running	<div></div>
Publicidade Radio	Running	<div></div>
Publicidade Dispositivos Moveis	Running	<div></div>

1

of 1

3

3 Records

Figura 5.11: Tabelas para Gestão de Contas e Campanhas

List of Users With Access To Publicidade Automóvel 3	
Client	Type Of Access
baldasman@gmail.com	Guest
<div> <div> <div>1</div> <div>of 1</div> </div> <div> <div>10</div> <div>1 Records</div> </div> </div>	
List Of Permissions For baldasman@gmail.com	
Campaign	Type Of Access
Golf VI	Administrator
<div> <div> <div>1</div> <div>of 1</div> </div> <div> <div>3</div> <div>1 Records</div> </div> </div>	

Add Record

Campaign:

Golf VI

Type Of Access:

Administrator

Administrator

Guest

None

Figura 5.12: Tabelas para Gestão de Permissões

Billing

From: 2009-01-21 To: 2009-06-28 Refresh

March 2009

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Date	End Date	Total	Details
-01	2009-06-30	86527.66 €	View Details
-01	2009-05-31	86089.59 €	View Details
-01	2009-04-30	88028.03 €	View Details
-01	2009-03-31	88068.07 €	View Details
-01	2009-02-28	82178.82 €	View Details
-01	2009-01-31	86269.22 €	View Details
-01	2008-12-31	85824.57 €	View Details
-01	2008-11-30	87252.73 €	View Details

Figura 5.13: Tabelas para Visualizar Facturas

Cada servidor tem uma maneira própria de construir os seus relatórios, tornou-se então necessário implementar um interpretador de relatórios capaz de extrair de um relatório toda a informação necessária à criação das campanhas no nosso sistema. Após a descodificação da informação contida no ficheiro, esta é mostrada ao utilizador, em forma de tabela, para que este proceda a uma validação final.

Deste modo o utilizador pode importar todas as suas campanhas em dois pequenos passos: criar o relatório no seu servidor e importá-lo na nossa aplicação.

5.4.3.4 Visualização de Facturas

Uma factura é constituída por um cabeçalho e um corpo. Para proporcionar uma vista geral das facturas sem ter de mostrar todos os detalhes de uma factura, pois isso iria ocupar grande parte da página reduzindo o número de facturas visíveis, optou-se por implementar um sistema de visualização com duas fases.

Numa primeira fase o utilizador pode visualizar todas as facturas emitidas num determinado intervalo de tempo, previamente seleccionado através de dois selectores presentes no topo da página. Estes selectores são interactivos, ou seja, o utilizador pode introduzir manualmente as datas, ou pode seleccionar as datas a partir de um calendário implementado com *Javascript* como se mostra na Figura 5.13. Depois o utilizador pode abrir cada uma das facturas, para assim consultar todos os detalhes da mesma. Desta maneira, consegue-se mostrar mais informação no mesmo espaço, tornando a interface mais simples e fácil de navegar.

5.4.4 Interface de Análise

Esta é uma das partes mais importantes do trabalho e à qual se dedicou mais tempo. De seguida vamos descrever algumas das soluções encontradas para implementar uma interface muito interactiva, mas ao mesmo tempo, muito simples de utilizar.

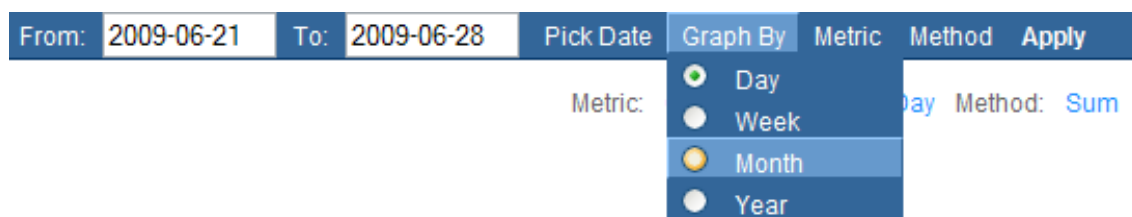


Figura 5.14: Menu para personalizar gráficos e tabelas

5.4.4.1 Dashboard de Campanhas

Esta zona da interface foi criada para oferecer ao utilizador uma vista geral sobre o desempenho de todas as suas campanhas. Para oferecer essa vista geral optou-se por usar um gráfico e uma tabela, ou seja, a mesma informação é representada de duas maneiras diferentes. De modo a permitir manipular facilmente a informação representada, tanto no gráfico como na tabela, foi criado um menu com efeito *drop-down* como se mostra na Figura 5.14. A partir deste menu o utilizador pode escolher as datas, métrica, método e escala temporal a utilizar na representação da informação. Um pequeno resumo das opções escolhidas está sempre disponível por baixo do menu.

Toda a informação disponível pode ser filtrada por campanha e data. Para isso foi criado um selector de campanhas onde o utilizador pode facilmente escolher quais as campanhas que deseja analisar através de filtros existentes ou escolhendo uma a uma, através de uma lista de campanhas disponível como se mostra na Figura 5.15. Devido às dimensões deste selector, optou-se por dar ao utilizador a opção de esconder este selector sempre que ele não está a ser utilizado, deixando assim mais espaço para o gráfico e tabela.

Para escolher o intervalo de datas desejado, criaram-se dois selectores onde o utilizador pode introduzir as datas manualmente ou utilizar o calendário disponibilizado para o fazer mais facilmente. Também foram criados selectores rápidos com as seguintes opções: hoje, esta semana, este mês e este ano. Dentro de cada intervalo temporal o utilizador pode ainda escolher a resolução da escala temporal: dia, semana, mês ou ano.

Muitas vezes os utilizadores estão interessados em analisar não só o comportamento individual de uma campanha mas também o desempenho de um conjunto de campanhas com alguma relação entre si. Para responder a este problema foram criados dois métodos diferentes de análise:



Figura 5.15: Selector de Campanhas

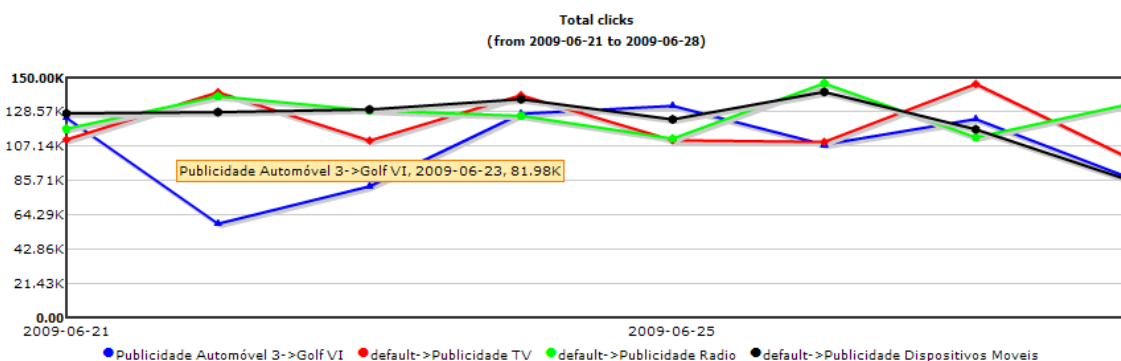


Figura 5.16: Gráfico com Método Comparativo

comparativo e somatório. No método comparativo, o utilizador após escolher as várias campanhas que deseja analisar, pode visualizar no gráfico as diferentes análises, representadas com diferentes cores, como se mostra na Figura 5.16. Deste modo pode comparar o desempenho individual de cada campanha. Se optar pelo método somatório, é representado apenas um gráfico com a soma de todas as campanhas seleccionadas como se mostra na Figura 5.17.

Toda a informação que é representada nos gráficos está também disponível em forma de tabela. No método comparativo, as tabelas mostram a informação relativa a cada uma das campanhas seleccionadas. Para alternar entre as diferentes campanhas foi colocado um selector no topo da tabela como se mostra na Figura 5.18. Toda a informação representada nas tabelas pode facilmente ser exportada para diferentes formatos de ficheiro utilizando os botões disponíveis no fundo da tabela.

Toda a informação necessária para desenhar os gráficos e preencher as tabelas é carregada utilizando AJAX para evitar ter de recarregar toda a página, assim apenas o gráfico ou a tabela são recarregados, diminuindo o tempo de resposta e aumentando desempenho da interface. Sempre que um utilizador efectua uma alteração nas opções de visualização, existe um botão no menu que muda de cor, chamando à atenção do utilizador para as alterações efectuadas como se mostra na

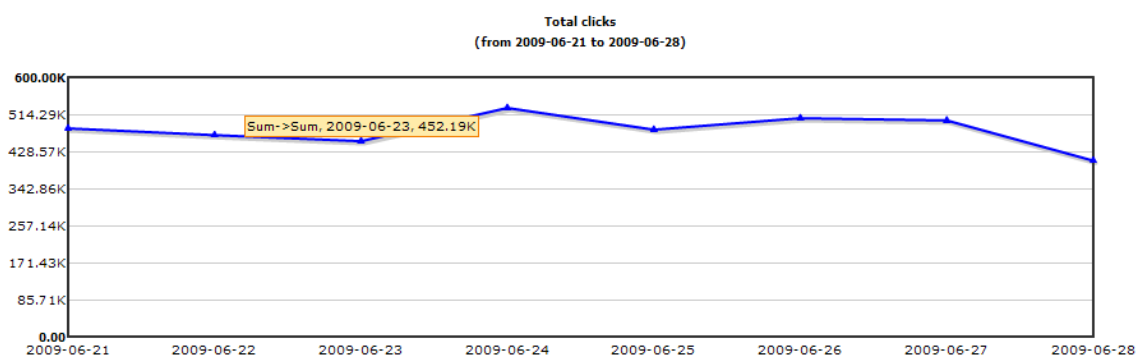


Figura 5.17: Gráfico com Método Somatório

Golf VI			
Date ▲	Account	Campaign	Total Clicks
2009-06-21 00:00:00+00	Publicidade Automóvel 3	Golf VI	124823
2009-06-22 00:00:00+00	Publicidade Automóvel 3	Golf VI	58745
2009-06-23 00:00:00+00	Publicidade Automóvel 3	Golf VI	81981
2009-06-24 00:00:00+00	Publicidade Automóvel 3	Golf VI	127479

Figura 5.18: Selector de Series

Figura 5.19. As novas opções só entram em vigor após a validação do utilizador.

5.4.4.2 Dashboard de Relatórios

Esta zona da interface foi pensada para dar ao utilizador uma vista geral de todas as análises disponibilizadas. Isto foi conseguido criando diversos *widgets*, cada um com um gráfico em tamanho reduzido da análise em questão, como se mostra na Figura 5.20. O utilizador pode mover, minimizar, maximizar, adicionar ou remover estes *widgets* a qualquer altura. Toda esta configuração fica guardada na base de dados para que, quando o utilizador voltar a utilizar esta aplicação, os *widgets* apareçam exactamente no mesmo sítio.

Para garantir que não existe sobreposição, foram criadas 3 colunas nas quais os *widgets* podem ser colocados. Para mover os *widgets* de uma coluna para outra, o utilizador só tem de os arrastar como se mostra na Figura 5.21, e largá-lo na posição desejada.

Para adicionar ou remover *widgets*, assim como guardar as configurações, o utilizador tem de utilizar a janela de configuração de *widgets*, como se mostra na Figura 5.22. Só é permitida uma instância de cada *widget*. Esta janela de configuração foi criada tirando partido de um efeito conhecido como *overlay* [54]. Isto consiste em criar uma janela por cima de outra, bloqueando toda a interacção com a janela que está por baixo, mas permitindo ver o que está a acontecer. Deste modo o utilizador pode adicionar os *widgets* e, ao mesmo tempo, ver onde eles estão a ser colocados.

Nesta zona da aplicação também está disponível o selector de campanhas e o selector de datas, descritos anteriormente, para filtrar a informação apresentada nos diferentes gráficos. Todos estes efeitos visuais foram implementados utilizando as diversas funcionalidades disponibilizadas pela *framework* de *Javascript* utilizada — *jQuery*.

From:	2009-06-21	To:	2009-06-28	Pick Date	Graph By	Metric	Method	Apply ?
Metric: Total Clicks Graph by: Week Method: Compare								

Figura 5.19: Botão de Validação

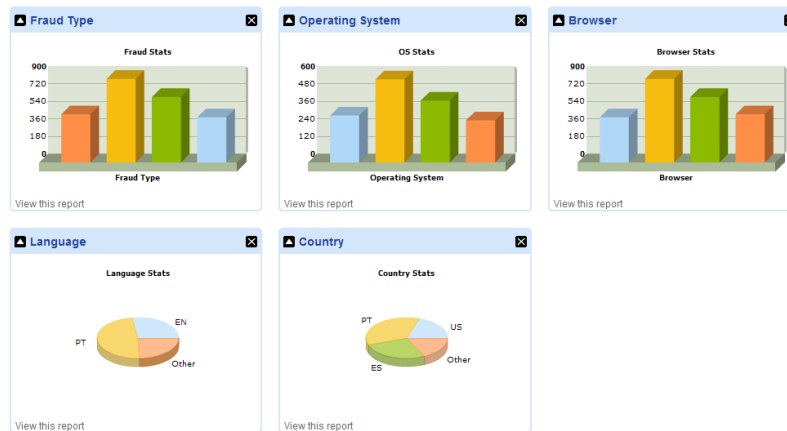


Figura 5.20: Dashboard Relatórios

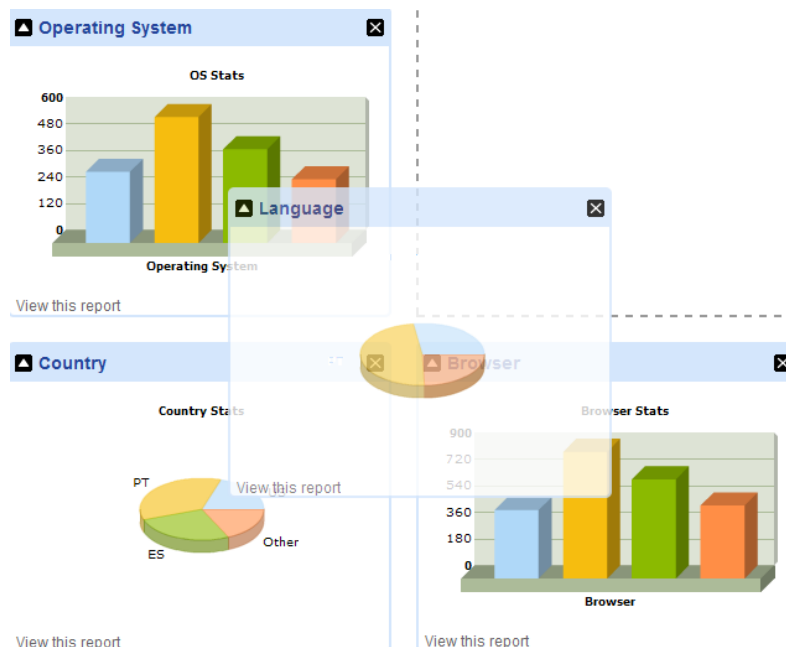


Figura 5.21: Mover Widgets

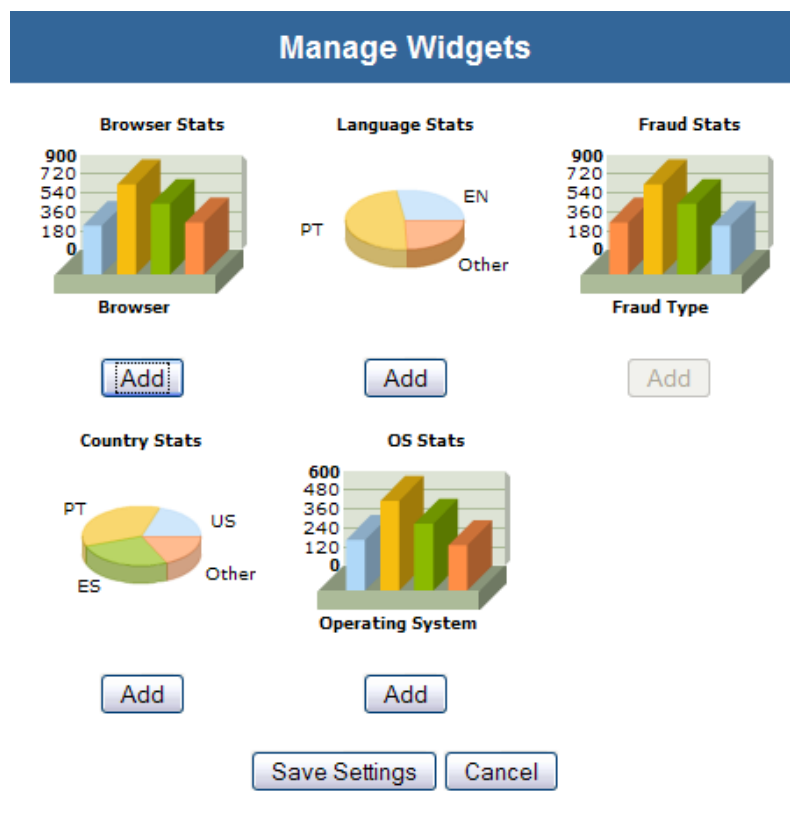


Figura 5.22: Configuração de Widgets

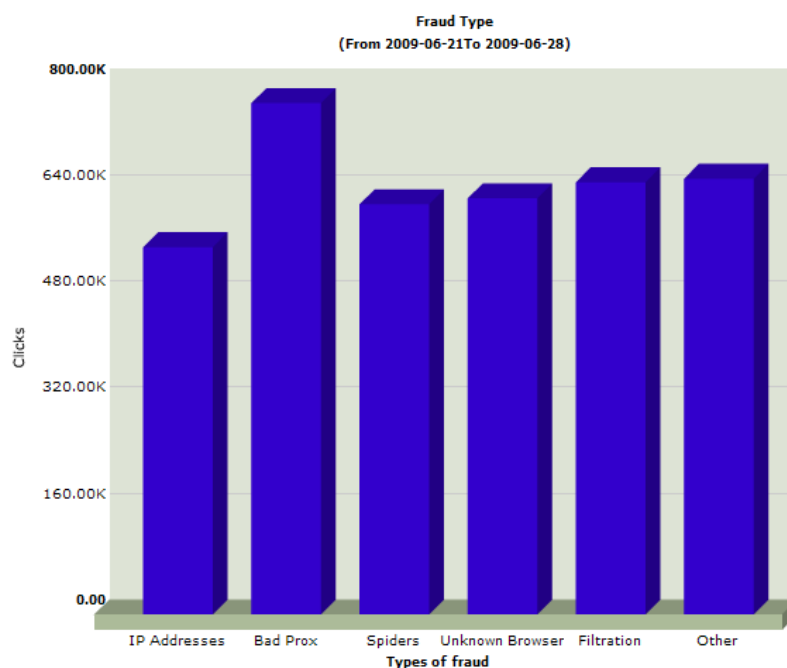


Figura 5.23: Gráfico de Barras — Modo Soma

5.4.4.3 Visualização de Relatórios

Existem muitas métricas que podem ser analisadas no contexto de auditoria *Web*. Para este projecto ficou decidido usar 5 tipos de análise diferente: *browser* utilizado pelo visitante, *sistema operativo* utilizado pelo visitante, país do visitante, língua do visitante e tipos de fraude.

Para cada uma das análise decidiu-se seguir a ideia implementada no *dashboard* de campanhas — usar um gráfico e uma tabela. Todas as funcionalidades disponíveis no *dashboard* de campanhas, também estão disponíveis aqui e, toda a sua configuração é transportada de uma funcionalidade para a outra através do uso de *cookies* ³. Desta forma, o utilizador apenas tem de configurar os filtros uma vez, visto que a sua configuração é guardada no *browser*, enquanto a sua sessão estiver activa.

Cada tipo de análise tem características próprias, por isso foram usados diferentes tipos de gráficos para representar cada uma das análises, como se mostra nas Figuras 5.23, 5.24 e 5.25.

5.4.5 Sistema de Ajuda

Um dos objectivos deste trabalho era integrar um sistema de ajuda. Ficou decidido, por parte da AuditMark, que a ferramenta a usar é o phpMyFAQ [55]. Esta uma ferramenta oferece as seguintes características:

³Pequeno texto armazenado no computador de um utilizador por um browser.

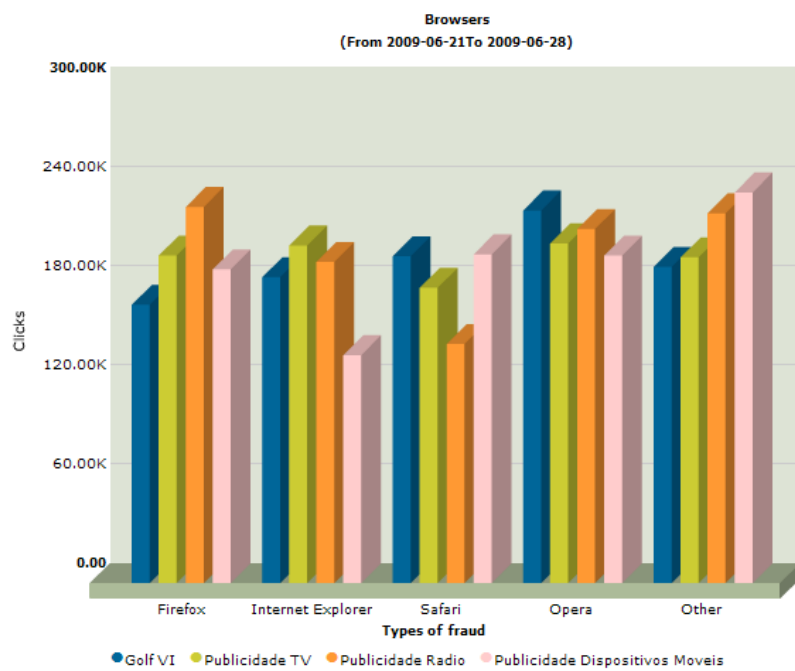


Figura 5.24: Gráfico de Barras — Modo Comparativo

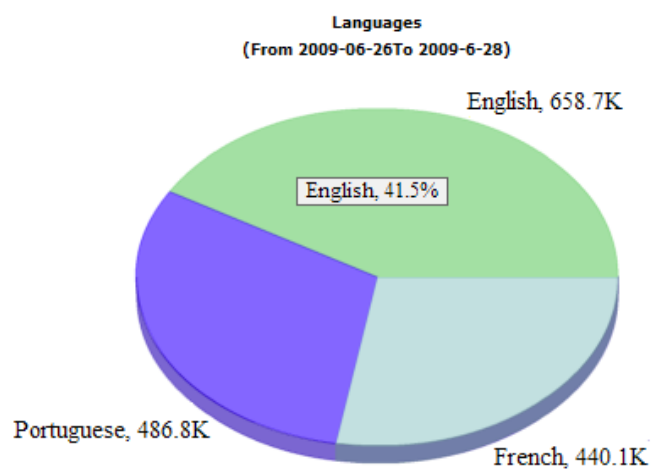


Figura 5.25: Gráfico Tarte

- Vários sistemas de base de dados suportados — MySQL (libmysql e mysqlnd), PostgreSQL, SQLite, Sybase, MS SQL Server, IBM DB2, IBM Cloudscape, Apache Derby, Oracle, Interbase ou Firebird;
- Sistema de Gestão de Conteúdos — Para gerir os utilizadores, grupos, notícias, categorias e comentários;
- Revisão do Sistema — Guarda as versões antigas como revisões, para que se possa voltar a versões antigas;
- Personalizável — Pode-se facilmente personalizar o layout, usando XHTML e CSS;
- Multi-língua — Suporta mais de 35 idiomas, incluindo o árabe, farsi e hebraico;
- FAQs mais populares — A página inicial mostra uma lista com as pesquisas mais populares;
- Exportação — Permite exportar a informação em 4 formatos diferentes: XHTML, PDF, XML e DocBook XML;
- Protecção contra spam — Utiliza *captchas* e listas de *IP's* proibidos para evitar spam.

Para completar o sistema de ajuda, foi apenas necessário personalizar o seu *layout* para ficar a condizer com o *layout* utilizado no resto da interface e introduzir uma série de artigos de ajuda. Estes artigos podem ser consultados directamente a partir da pesquisa disponibilizada no sistema de ajuda ou a partir dos *links* disponíveis no menu lateral da interface.

5.5 Testes

Uma parte importante no desenvolvimento de qualquer aplicação, são os testes realizados para avaliar o trabalho realizado. Para avaliar este trabalho foram realizados os seguintes testes:

- Testes de Implementação — Validação de código HTML e CSS;
- Testes de Segurança;
- Testes de Validação — Validação por parte do cliente;

Nos parágrafos seguintes vamos descrever e analisar os resultados obtidos em cada um dos testes.

5.5.1 Teste de Implementação

Para garantir uma correcta implementação utilizaram-se as seguintes ferramentas disponibilizadas pelo W3C [56]: CSS Validation Service [57] e Markup Validation Service [58]. O resultados destes testes demonstrou que se desenvolveu uma aplicação que respeita todas as regras para desenvolver código HTML e CSS.

5.5.2 Testes de Segurança

Para detectar vulnerabilidades na aplicação, foi utilizada a seguinte ferramenta — Paros [59]. Esta ferramenta permite identificar, entre outros, os seguintes problemas:

- Ficheiros obsoletos;
- Divulgação de *IP*'s privados;
- Possibilidade de reescrever o *ID* da sessão no URL;
- Ferramenta para completar passwords activa no *browser*;
- Possibilidade de listar directórios;
- Injecção de SQL;
- Injecção de CRLF ⁴;

Após a conclusão deste teste encontrou-se algumas vulnerabilidades que foram prontamente resolvidas. Na Tabela 5.1 podem ser consultadas as vulnerabilidades encontradas e a respectiva solução.

Tabela 5.1: Vulnerabilidades encontradas e respectiva solução

Vulnerabilidade	Solução
Possibilidade de listar directórios	Desactivar a respectiva opção no servidor Apache.
Ferramenta para completar passwords activa no <i>browser</i>	Desactivar esta opção adicionando a opção <code>AUTOCOMPLETE='OFF'</code> aos formulários.
Ficheiros obsoletos	Apagar os respectivos ficheiros.

5.5.3 Teste de Validação

Este é provavelmente o teste mais importante, pois para podermos afirmar que atingimos todos os objectivos a que nos propusemos, temos de obter a validação por parte do cliente. Para demonstrar o correcto funcionamento de todas as funcionalidades ao cliente, foi realizado o seguinte conjunto de testes:

- Aceder ao sítio *Web* e enviar um email para a AuditMark;
- Efectuar uma pesquisa dentro do sítio *Web*;
- Mudar a língua do sítio *Web*;

⁴Carriage Return and Line Feed

- Aceder à Área de Cliente e efectuar o registo;
- Aceder à Área de Cliente e efectuar o registo usando uma identidade do OpenId;
- Recuperar a password;
- Recuperar o username;
- Efectuar *login* utilizando o *username* e *password*;
- Editar as configurações pessoais;
- Criar uma nova conta;
- Criar uma nova campanha dentro da conta criada;
- Importar campanhas a partir de um ficheiro previamente gerado no Google Analytics;
- Adicionar um utilizador à conta criada;
- Editar as permissões do novo utilizador;
- Visualizar os dados de uma factura;
- Visualizar o *dashboard* de campanhas, seleccionar um intervalo de tempo diferente e seleccionar várias campanhas;
- Visualizar o *dashboard* de relatórios, mover, adicionar e remover um *widget*;
- Visualizar um relatório e exportar a informação em formato CSV;
- Aceder ao Sistema de Ajuda e realizar uma pesquisa;

Todos os testes foram bem sucedidos, pelo que podemos concluir que atingimos os objectivos propostos para este projecto.

Capítulo 6

Conclusões e Trabalho Futuro

O trabalho proposto tinha como objectivo criar uma aplicação para gestão de uma plataforma de auditoria *Web*, com uma interface interactiva e simples de usar. Para isso exploraram-se as soluções semelhantes já existentes, bem como todas as tecnologias e ferramentas usadas em desenvolvimento *Web*.

A análise de soluções semelhantes permitiu compreender as necessidades do ponto de vista do utilizador, para assim desenvolvermos um conjunto de funcionalidades mais completo e proporcionar ao utilizador mais do que uma simples análise, uma visão personalizada e compreensível do problema. O conhecimento de todas as tecnologias disponíveis, permitiu desenvolver uma aplicação robusta, dinâmica, intuitiva e, sobretudo, fácil de utilizar.

O uso de uma *framework* de apoio ao desenvolvimento revelou-se muito importante não só na organização e estruturação do código, através do uso do modelo MVC, mas também devido às funcionalidades disponibilizadas tais como: motor de busca, métodos de criação e personalização de formulários, ambiente multi-língua, integração com o OpenId, etc; o que permitiu poupar muito tempo de desenvolvimento. A *framework* de *Javascript* utilizada revelou ser uma óptima escolha, pois não só disponibiliza uma grande quantidade de efeitos visuais que permitiram tornar a interface muito interactiva, como é extremamente simples de usar. O uso da *framework* jQuery também tornou possível o uso da *mini-framework* jqGrid, que foi muito importante na criação e manipulação de todas as tabelas.

A utilização de um sistema de gráficos baseado em Flash e controlável através de Javascript provou ser uma escolha acertada, pois conseguiu-se criar uma série de gráficos visualmente agradáveis, interactivos e actualizados através de AJAX. Isto foi decisivo para diminuir o tempo resposta e aumentar o desempenho da aplicação.

Os testes realizados confirmam que todos os objectivos foram atingidos, porém para que esta ferramenta se torne numa séria alternativa às soluções existentes ainda há algum trabalho a realizar. Algumas das sugestões para um trabalho futuro são:

- Incluir mais tipos de análise;

- Adquirir uma licença profissional para o sistema de gráficos, para aumentar o número de gráficos e efeitos visuais disponíveis;
- Permitir a exportação de informação em mais formatos, como por exemplo ODF;
- Incluir mais componentes em Flash, nomeadamente no sítio *Web* da empresa, para assim o tornar mais dinâmico;
- Permitir a importação de campanhas a partir de todos os servidores de anúncios existentes;
- Permitir a exportação de gráficos.

Referências

- [1] North America Equity Research. North america equity research, Janeiro 2009. https://mm.jpmorgan.com/stp/t/c.do?i=2082C-248&u=a_p*d_170762.pdf*h_-3ohpnmv. Acedido pela última vez em 5 de Fevereiro de 2009.
- [2] Google. Google analytics, Junho 2009. <http://www.google.com/analytics>. Acedido pela última vez em 26 de Junho de 2009.
- [3] Webtrends. Webtrends your trusted analytics advisor, Junho 2009. <http://www.webtrends.com>. Acedido pela última vez em 26 de Junho de 2009.
- [4] Omniture. Online business optimization by omniture, Junho 2009. <http://www.omniture.com>. Acedido pela última vez em 26 de Junho de 2009.
- [5] ClickTracks. Web analytics for web marketing analysis by clicktracks analytics, Junho 2009. <http://www.clicktracks.com>. Acedido pela última vez em 26 de Junho de 2009.
- [6] NetTracker. Unica - unica nettracker, Junho 2009. <http://netinsight.unica.com/NetTracker.htm>. Acedido pela última vez em 26 de Junho de 2009.
- [7] Google. Urchin software from google, Junho 2009. <http://www.google.com/urchin/features.html>. Acedido pela última vez em 26 de Junho de 2009.
- [8] Click Forensics. Click forensics, Junho 2009. <http://www.clickforensics.com>. Acedido pela última vez em 26 de Junho de 2009.
- [9] Bill Kennedy Chuck Musciano. *HTML: The Definitive Guide*. O'Reilly, 3 edition, 2009.
- [10] Erik T. Ray. *Learning XML*. O'Reilly, 2 edition, 2003.
- [11] Eric Meyer. *Cascading Style Sheets: The Definitive Guide*. O'Reilly, 3 edition, 2009.
- [12] David Flanagan. *Javascript: The Definitive Guide*. O'Reilly, 3 edition, 1998.
- [13] W3C. Extensible markup language (xml), Outubro 2008. <http://www.w3.org/XML/>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [14] Cascading style sheets, Janeiro 2009. <http://www.w3.org/Style/CSS>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [15] Ajax patterns, Janeiro 2009. <http://ajaxpatterns.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [16] MySQL. Mysql: The world's most popular open source database, Junho 2009. <http://www.mysql.com>. Acedido pela última vez em 26 de Junho de 2009.

- [17] Oracle. Oracle database lite, Junho 2009. <http://www.oracle.com/technology/products/lite/index.html>. Acedido pela última vez em 26 de Junho de 2009.
- [18] PostgreSQL. Postgresql: The world's most advanced open source database, Junho 2009. <http://www.postgresql.org/>. Acedido pela última vez em 26 de Junho de 2009.
- [19] jQuery. jquery: The write less, do more, javascript library, Junho 2009. <http://jquery.com>. Acedido pela última vez em 26 de Junho de 2009.
- [20] MooTools. Mootools - a compact javascript framework, Junho 2009. <http://mootools.net>. Acedido pela última vez em 26 de Junho de 2009.
- [21] Prototype. Prototype javascript framework: Easy ajax and dom manipulation for dynamic web applications, Junho 2009. <http://www.prototypejs.org>. Acedido pela última vez em 26 de Junho de 2009.
- [22] Karl Swedberg Hasin Hayder, Jonathan Chaffer. *Learning jQuery 1.3*. PACKT Publishing, 1 edition, 2009.
- [23] AnyChart. Anychart, Junho 2009. <http://www.anychart.com>. Acedido pela última vez em 26 de Junho de 2009.
- [24] Flash Charts Pro. Flash charts pro, Junho 2009. <http://www.web-site-scripts.com/flash-charts/index.html>. Acedido pela última vez em 26 de Junho de 2009.
- [25] amCharts. amcharts, Junho 2009. <http://www.amcharts.com>. Acedido pela última vez em 26 de Junho de 2009.
- [26] Fusion Charts. Fusion charts, Junho 2009. <http://www.fusioncharts.com>. Acedido pela última vez em 26 de Junho de 2009.
- [27] Emprise Java Script Charts. Emprise java script charts, Junho 2009. <http://www.ejschart.com/index.php>. Acedido pela última vez em 26 de Junho de 2009.
- [28] Swiff Chart. Swiff chart, Junho 2009. <http://www.globfx.com/products/swfchart/features.php>. Acedido pela última vez em 26 de Junho de 2009.
- [29] XML/SWF. Xml/swf, Junho 2009. http://www.maani.us/xml_charts/index.php. Acedido pela última vez em 26 de Junho de 2009.
- [30] Mohamed Fayad and Douglas Schmidt. Object-oriented application frameworks. *Communications of the ACM*, 40(10):32–38, 1997.
- [31] Peter MacIntyre Kevin Tatroe, Rasmus Lerdorf. *Programming PHP*. O'Reilly, 2 edition, 2006.
- [32] Akelos PHP Framework. Akelos php framework, Janeiro 2009. <http://www.akelos.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [33] ash.MVC: Overview. ash.mvc: Overview, Janeiro 2009. <http://ash-mvc.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [34] CakePHP. Cakephp: the rapid development php framework., Janeiro 2009. <http://cakephp.org>. Acedido pela última vez em 5 de Fevereiro de 2009.

- [35] Codeigniter. Codeigniter, Janeiro 2009. <http://codeigniter.com>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [36] DIY. Diy, Janeiro 2009. <http://www.diynetwork.com>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [37] eZ Components. ez components, Janeiro 2009. <http://ez.no/ezcomponents>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [38] Fusebox. Fusebox, Janeiro 2009. <http://www.fusebox.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [39] TRAX. Php on trax, Janeiro 2009. <http://www.phpontrax.com>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [40] PHPDevShell. Phpdevshell, Janeiro 2009. <http://www.phpdevshell.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [41] PhpOpenBiz. Phpopenbiz, Janeiro 2009. <http://www.phpopenbiz.org/jim>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [42] Prado. Prado, Janeiro 2009. <http://www.xisc.com>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [43] QPHP. Qphp, Janeiro 2009. <http://qphp.net>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [44] Seagull. Seagull, Janeiro 2009. <http://seagullproject.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [45] Symfony. Symfony, Janeiro 2009. <http://www.symfony-project.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [46] WACT. Wact, Janeiro 2009. <http://www.phpwact.org>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [47] WASP. Wasp, Janeiro 2009. <http://wasp.sourceforge.net/content>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [48] Yii. Yii, Janeiro 2009. <http://www.yiiframework.com>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [49] Zend. Zend, Janeiro 2009. <http://www.zend.com/en>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [50] ZooP. Zoop, Janeiro 2009. <http://zoopframework.com>. Acedido pela última vez em 5 de Fevereiro de 2009.
- [51] Marco Tabini John Coggeshall. *The Zend PHP Certification Practice Test Book*. php | architect, 1 edition, 2005.
- [52] Alberto Manuel Rodrigues da Silva and Carlos Alberto Escalera Videira. *UML, Metodologias e Ferramentas*, volume I. CENTROATLANTICO.PT, 2 edition, 2005.

- [53] Serena. Serena software - application development & business software solutions, Junho 2009. <http://www.serena.com>. Acedido pela última vez em 26 de Junho de 2009.
- [54] jQuery. jquery tools / overlay - overlay your html with eyecandy, Junho 2009. <http://flowplayer.org/tools/overlay.html>. Acedido pela última vez em 26 de Junho de 2009.
- [55] phpMyFAQ. phpmyfaq homepage - open source faq system for php and mysql, postgresql and other databases, Junho 2009. <http://www.phpmyfaq.de>. Acedido pela última vez em 26 de Junho de 2009.
- [56] W3C. World wide web consortium - web standards, Junho 2009. <http://www.w3.org>. Acedido pela última vez em 26 de Junho de 2009.
- [57] W3C. Serviço de validação de css do w3c, Junho 2009. <http://www.phpmyfaq.de>. Acedido pela última vez em 26 de Junho de 2009.
- [58] W3C. The w3c markup validation service, Junho 2009. <http://validator.w3.org>. Acedido pela última vez em 26 de Junho de 2009.
- [59] Paros. Parosproxy.org - web application security, Junho 2009. <http://www.parosproxy.org/index.shtml>. Acedido pela última vez em 26 de Junho de 2009.

Anexo A

Análise dos Servidores de Anúncios

Para permitir importar informação de servidores de anúncios foi necessário perceber quais as informações que estes disponibilizam. Na análise realizada foram considerados os seguintes servidores: OpenX, Google AdWords, Yahoo Ads e Microsoft AdCenter. Esta escolha deve-se ao facto de actualmente estes serem os servidores mais utilizados pelos utilizadores.

Após uma breve análise concluiu-se que, através da exportação de um relatório, os servidores de anúncios disponibilizam informações sobre os seguintes elementos: conta, campanha, grupo de anúncios e anúncio. Nas Tabelas [A.1](#), [A.2](#), [A](#) e [A.4](#) são apresentados atributos de cada um desses elementos.

Como se pode comprovar nas tabelas apresentadas, cada servidor caracteriza os diferentes elementos de uma forma distinta, sendo muito difícil encontrar pontos em comum entre eles. Devido a esta dificuldade, no processo de importação apenas vamos considerar a informação relativa às campanhas. Deste modo perde-se alguma informação sobre a organização das campanhas mas garante-se a compatibilidade com os vários servidores.

Tabela A.1: Atributos da Conta

OpenX	Ad Words	Yahoo Ads	Ad Center
name	descriptiveName	name	Account Name
	customerId	ID	Account Id
			Account Number
			Agency Contact Name
	currencyCode		Preferred Currency Type
language	languagePreference		Preferred Language Type
			Status
		displayURL	
		fiscalCode	
timezone			
contact	billingAddress		

Tabela A.2: Atributos da Campanha

OpenX	Ad Words	Yahoo Ads	Ad Center
id	id	ID	Id
name	name	name	Name
startDay	startDay	startDate	
endDay	endDay	endDate	
	languageTargeting		
status	status	status	Status
		accountID	
		description	Description
			Time Zone

Tabela A.3: Atributos do Grupo de Anúncios

OpenX	Ad Words	Yahoo Ads	Ad Center
	campaignId	campaignID	
	id	ID	Id
	name	name	Name
	status	status	Status
		accountID	

Tabela A.4: Atributos do Anúncio

	adGroupId	adGroupID	
Type	adType		Type
destination URL	destinationUrl	url	Destination Url
description	description	description	
	displayUrl	displayUrl	Display Url
	id	ID	Id
	status	status	Status
keywords			
		accountID	
		name	
		title	Title
		Editorial Status	Editorial Status